

REVIEW

Open Access



GLDPC-Staircase AL-FEC codes: a fundamental study and new results

Ferdaouss Mattoussi^{1*}, Vincent Roca¹ and Bessem Sayadi²

Abstract

This paper provides fundamentals in the design and analysis of Generalized Low-Density Parity Check (GLDPC)-Staircase codes over the erasure channel. These codes are constructed by extending an LDPC-Staircase code (base code) using Reed-Solomon (RS) codes (outer codes) in order to benefit from more powerful decoders. The GLDPC-Staircase coding scheme adds, in addition to the LDPC-Staircase repair symbols, extra-repair symbols that can be produced on demand and in large quantities, which provides small rate capabilities. Therefore, these codes are extremely flexible as they can be tuned to behave either like predefined rate LDPC-Staircase codes at one extreme, or like a single RS code at another extreme, or like small rate codes. Concerning the code design, we show that RS codes with “quasi” Hankel matrix-based construction fulfill the desired structure properties, and that a hybrid (IT/RS/ML) decoding is feasible that achieves maximum likelihood (ML) correction capabilities at a lower complexity. Concerning performance analysis, we detail an asymptotic analysis method based on density evolution (DE), extrinsic information transfer (EXIT), and the area theorem. Based on several asymptotic and finite length results, after selecting the optimal internal parameters, we demonstrate that GLDPC-Staircase codes feature excellent erasure recovery capabilities, close to that of ideal codes, both with large and very small objects. From this point of view, they outperform LDPC-Staircase and Raptor codes and achieve correction capabilities close to those of RaptorQ codes. Therefore, all these results make GLDPC-Staircase codes a universal Application-Layer FEC (AL-FEC) solution for many situations that require erasure protection such as media streaming or file multicast transmission.

Keywords: AL-FEC codes, GLDPC codes, LDPC-Staircase codes, ML decoding, DE and EXIT tools, Erasure channels

1 Review

Low-Density Parity Check (LDPC) codes have been intensively studied during the last decade due to their near-Shannon limit performance under iterative belief-propagation (BP) decoding [1–3]. A (N, K) LDPC code, where N is the code length and K is its dimension, can be graphically represented as a bipartite graph with N “variable nodes” (VNs) and $M = N - K$ “check nodes” (CNs). Equivalently, LDPC codes can be represented through their H_L parity check matrix translating the connection between VNs and CNs. The degree of a VN or a CN is defined as the number of edges connected to it. A VN of degree n can be interpreted as a “length repetition code” $(n, 1)$, i.e., as a linear block code repeating n times its single information bit towards the CN set. Similarly, a CN

of degree n can be interpreted as a Single Parity Check (SPC) code $(n, n - 1)$, i.e., as a linear block code associated with one parity equation. To improve error floor, minimal distance, and decoding complexity performances, a generalization of these codes was suggested by Tanner in [3], for which subsets of the variable nodes obey a more complex constraint than an SPC constraint. The SPC check nodes in a GLDPC structure are replaced with a generic linear block codes (n, k) referred to as sub-codes or component codes while the sparse graph representation is kept unchanged. More powerful decoders at the check nodes have been investigated by several researchers in recent years after the work of Boutros et al. [4] and Lentmaier and Zigangirov [5] where BCH codes and Hamming codes were proposed as component codes, respectively. Later several works, on several channels, have been carried out in order to afford very large minimum distance and exhibit performance approaching Shannon’s limit. Each construction differs from others by modifying the

*Correspondence: mattoussi.ferdaouss@gmail.com

¹Inria, PRIVATICS research Team, Grenoble, France

Full list of author information is available at the end of the article

linear block codes (component codes) on the check nodes [6, 7, 7–11] or/and the distribution of the structure of GLDPC codes [6] to offer a good compromise between waterfall performance and error floor under iterative decoding.

A GLDPC-Staircase code is an LDPC-Staircase code [12] in which the constraint nodes of the code graph are Reed-Solomon (RS) codes (rather than SPCs) in order to benefit from more powerful decoders. The construction of these RS codes, with the desired properties, is omitted from the initial work [13]. Therefore in [14], we introduce RS code-based Hankel matrices to that purpose. GLDPC-Staircase codes differ from the GLDPC codes proposed by Tanner and their successive variants. In particular, the GLDPC-Staircase coding scheme allows each check node to produce a potentially large number of repair symbols in terms of RS codes, called *extra-repair symbols*, on demand. These extra-repair symbols extend the base LDPC-Staircase code and *very small rates* are easily achievable. This feature is well suited to situations where channel conditions can be worse than expected and to fountain-like content distribution applications. More generally, these codes can easily be tuned to behave either like predefined rate LDPC-Staircase codes at one extreme, or like a single RS code at another extreme, or like a small rate code.

From a decoding perspective, we propose a new *hybrid (IT/RS/ML) decoding* approach that achieves the optimal correction capabilities of ML decoding at a lower complexity [14]. Finally, in order to analyze their performance, we detail in [15] an asymptotic analysis method based on the density evolution (DE) and extrinsic information transfer (EXIT) tools and the area theorem. Then, using this theoretical analysis combined with a finite length analysis, we discuss the impacts of the code structure and its internal parameters on performance.

Asymptotic and finite length analyses show that these codes achieve excellent decoding performance (i.e., good average decoding overhead, good waterfall region, small error floor, and channel capacity approaching performance), close to that of ideal codes, both with very large and very small objects. This independence with respect to the code dimension is a key practical benefit (e.g., LDPC codes are known to be asymptotically good only). We show in this work that our codes outperform the Raptor codes as well as some GLDPC codes, while being close to RaptorQ codes. Their extreme flexibility makes it possible to tune them to perfectly match each use-case (like low bit-rate streaming applications or at the opposite large file multicast transmission). The purpose of this paper is to give the reader a detailed overview of GLDPC-Staircase codes and to provide new results.

This paper is organized as follows. Section 2 focuses on the design of GLDPC-Staircase codes based on RS codes.

Then in Section 3, we explain the proposed asymptotic analysis method. Section 4 presents several analyses and optimizations of GLDPC-Staircase codes. Then, we analyze the achieved performance, compare these codes with other erasure codes, and provide preliminary decoding complexity results in Section 5. Finally, we conclude.

2 GLDPC-Staircase code design

2.1 Code description

As mentioned in introduction, GLDPC-Staircase codes are constructed from:

- *LDPC-Staircase code*: this is the *base code* with length N_L and dimension K . Let $M_L = N_L - K$ and let $H_L = (H_1|H_2)$ be the associated parity check matrix¹. From the LDPC-Staircase viewpoint, each row of H_L defines the connections between the source and LDPC repair symbols. From the GLDPC-Staircase viewpoint, each row of H_L defines the connections between the RS repair symbols and the source and LDPC repair symbols. Consequently, each LDPC-Staircase CN is represented as a powerful CN, called *generalized check node*, with GLDPC-Staircase codes.
- *RS codes*: they are the *outer codes* (or components codes). A generalized check node of index m can generate $e(m)$ extra-repair symbols from the RS point of view (plus one LDPC repair symbol if we use scheme A as we will see below). This is done with an $RS(n_m, k_m)$ encoding over $GF(2^b)$ with $0 \leq e(m) \leq E$ and $m = 1, \dots, M_L$. Here, E , k_m , and n_m are respectively the maximum number of extra-repair symbols per generalized check node, the RS code dimension and length for the generalized check node m .

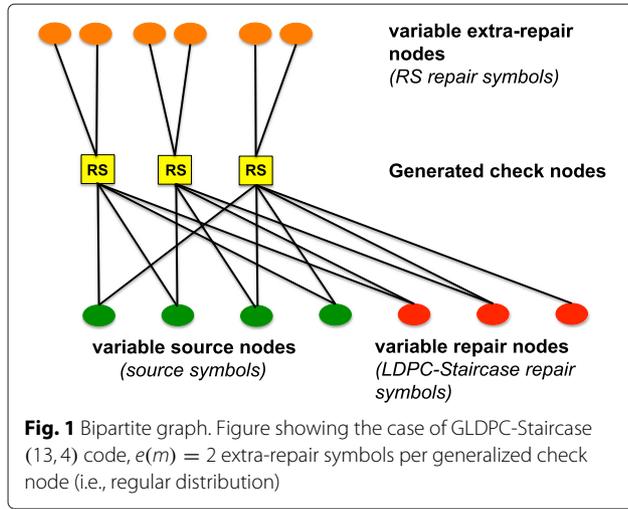
Figure 1 illustrates the bipartite graph of a GLDPC-Staircase (N_G, K) code of length N_G and dimension K . It is composed of two sets of nodes:

- *the generalized check node* that corresponds to RS codes;
- *the variable nodes* (VN) further divided into three categories:
 - source symbols;
 - LDPC repair symbols;
 - extra-repair symbols.

2.2 Schemes A and B

Let us now define two variants, schemes A and B, depending on the definition of n_m and k_m :

- *Scheme A*
For row $m > 1$, the source symbols (from the user viewpoint) involved in this row plus the LDPC repair symbol of row $m - 1$ are considered as source symbols from the RS viewpoint. The new LDPC repair symbol



plus the $e(m)$ extra-repair symbols are considered as repair symbols from the RS viewpoint. Therefore, the LDPC repair symbol is also an RS repair symbol. For $m = 1$, the only difference is that there is no previous repair symbol (beginning of the staircase). No matter the row, we have

$$n_m = k_m + 1 + e(m) \text{ and } k_m = d_r(m) - 1, \quad (1)$$

where $d_r(m)$ is the degree of row m of H_L . Due to the staircase structure of H_2 , it follows

$$d_r(m) = \begin{cases} d_{r_{H_1}} + 1 & \text{if } m = 1; \left(d_{r_{H_1}} = \frac{N_1}{r_L - 1} \right) \\ d_{r_{H_1}} + 2 & \text{if } m > 1; \left(d_{r_{H_1}} = \frac{N_1}{r_L - 1} \right). \end{cases} \quad (2)$$

In order to fulfill the duality property of the LDPC repair symbols, we propose in [16] a specific construction of RS codes based on “quasi” Hankel matrix. The generator matrix G of these codes has the following form:

$$G = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 & 1 & 1 & \dots & \dots & 1 \\ 0 & 1 & \ddots & \ddots & \vdots & 1 & b_1 & b_2 & \dots & b_{n_m - k_m} \\ 0 & 0 & \ddots & \ddots & \vdots & 1 & b_2 & \ddots & \ddots & b_{n_m - k_m + 1} \\ \vdots & \vdots & \ddots & \ddots & 0 & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 1 & b_{k_m} & \ddots & \ddots & b_{n_m - 1} \end{bmatrix} \quad (3)$$

where $b_i = \frac{1}{1 - y^i}$, for $1 \leq i \leq q - 1$, y is an arbitrary primitive element of $GF(q)$ and y^i is computed over $GF(q)$.

Thanks to the column full of “1” in G for the first RS repair symbol, this latter can also be considered as an LDPC-Staircase symbol (it is the XOR sum of source symbols from the RS viewpoint).

• Scheme B

For each row m , the various source symbols (from the user viewpoint) involved in this row plus the LDPC repair symbol(s) are considered as source symbols from the RS viewpoint. The $e(m)$ extra-repair symbols are the only repair symbols from the RS viewpoint. No matter the row, we have

$$n_m = k_m + e(m) \text{ and } k_m = d_r(m). \quad (4)$$

Here, any RS code (e.g., based on Hankel, Cauchy, or Vandermonde matrices) can be used.

2.3 Extra-repair symbol regular/irregular distributions

For a fixed code rate r_L of LDPC-Staircase code (N_L, K) , the code rate of the GLDPC-Staircase code is given by

$$r_G = \frac{K}{N_L + M_L \bar{f}} = \frac{r_L}{1 + (1 - r_L) \bar{f}}. \quad (5)$$

where \bar{f} is the average number of extra-repair symbols per generalized check node:

$$\bar{f} = \sum_{e=0}^E (f_e \cdot e), \quad (6)$$

and f_e denotes the fraction of generalized check nodes with e extra-repair symbols:

$$f_e = \frac{\text{card}\{m = 1 \dots M_L \mid e(m) = e\}}{M_L} \quad (7)$$

We can consider the following two distributions of extra-repair symbols on the various generalized check nodes:

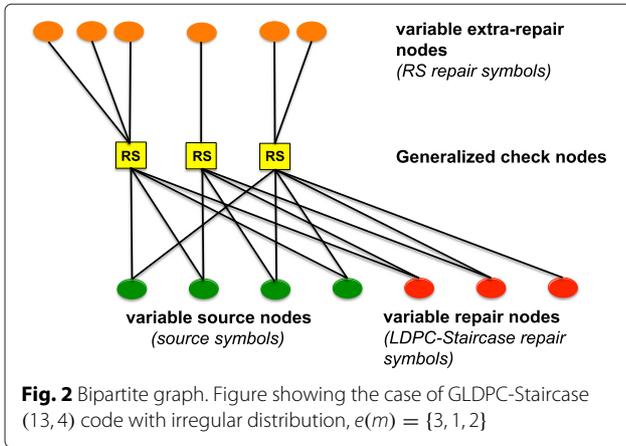
- Regular distribution: $f_e = 0$ for $e \in \{0, 1, \dots, E - 1\}$ and $f_E = 1$. Thus, each generalized check node m has the same number $e(m) = E$ of extra-repair symbols and the rate of the extended code (GLDPC-Staircase code) is

$$r_G = \frac{r_L}{1 + (1 - r_L) * E} \quad (8)$$

Figure 1 shows such a regular variant.

- Irregular distribution: the generalized check nodes can have a different number of extra-repair symbols. Figure 2 shows such an irregular variant. Cunche et al. [13] shows that there exists an irregular uniform distribution of extra-repair symbols which achieves performance close to the optimal irregular distribution. This irregular uniform distribution allows to allocate the extra-repair symbols with $\bar{f} = \frac{E}{2}$ and $f_e = \frac{1}{E+1}$ for $e \in \{0, 1, \dots, E\}$.

Throughout this paper, we only consider the regular distribution and the irregular uniform distribution, and we assess in Section 4.4.1 their impacts on performance.



2.4 Encoding method

Encoding generates two types of repair symbols:

- M_L LDPC-Staircase repair symbols, (p_1, \dots, p_{M_L}) , and
- $M_L f$ extra-repair symbols, $((e_{1,1}, \dots, e_{1,e(1)}), \dots, (e_{M_L,1}, \dots, e_{M_L,e(M_L)}))$.

Let $S = (S_1, S_2, \dots, S_K)$ be the K source symbols. The (p_1, \dots, p_{M_L}) repair symbols are computed following the “stairs” of H_L : p_m is the XOR sum of the subset x of S of source symbols that have a “1” coefficient in row m , plus p_{m-1} if $m > 1$.

Then, the $e(m)$ extra-repair symbols for row m are computed by multiplying the k_m LDPC symbols by the systematic generator matrix G_m of RS (n_m, k_m) associated to this row².

Example 2.1. Consider the GLDPC-Staircase code, scheme A, defined by the bipartite graph of Fig. 1. We have $N_G = 13$, $K = 4$ and exactly $e(m) = 2$ extra-repair symbols per generalized check node (regular distribution). H_L and the various RS codes are as follows:

$$H_L = \begin{array}{cccc|ccc} S_1 & S_2 & S_3 & S_4 & P_1 & P_2 & P_3 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 0 \\ 0 & 1 & 1 & 0 & \mathbf{1} & \mathbf{1} & 0 \\ 1 & 0 & 0 & 1 & 0 & \mathbf{1} & \mathbf{1} \end{array} \right] & \begin{array}{l} RS_1 = RS(6, 3) \\ RS_2 = RS(6, 3) \\ RS_3 = RS(6, 3) \end{array} \end{array} \quad (9)$$

We note that here (regular distribution and scheme A), the same RS code can be used for all the rows. Its generator matrix G_{rs} has the form

$$G_{rs} = \begin{array}{cccccc} & & & G_{rs}^4 & G_{rs}^5 & G_{rs}^6 \\ & & & \downarrow & \downarrow & \downarrow \\ \left[\begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & b_1 & b_2 \\ 0 & 0 & 1 & 1 & b_2 & b_2 \end{array} \right] & (10) \end{array}$$

To summarize, encoding is as follows:

- First row, using $x = (S_1, S_2, S_4)$, produces

$$P_1 = G_{rs}^4 \times (S_1, S_2, S_4)$$

$$e_{1,1} = G_{rs}^5 \times (S_1, S_2, S_4)$$

$$e_{1,2} = G_{rs}^6 \times (S_1, S_2, S_4)$$

- Second row, using $x = (S_2, S_3)$, produces

$$P_2 = G_{rs}^4 \times (S_2, S_3, P_1)$$

$$e_{2,1} = G_{rs}^5 \times (S_2, S_3, P_1)$$

$$e_{2,2} = G_{rs}^6 \times (S_2, S_3, S_1)$$

- Third row, using $x = (S_1, S_4)$, produces

$$P_3 = G_{rs}^4 \times (S_1, S_4, P_2)$$

$$e_{3,1} = G_{rs}^5 \times (S_1, S_4, P_2)$$

$$e_{3,2} = G_{rs}^6 \times (S_1, S_4, P_2)$$

A key advantage is the fact that extra-repair symbols can be produced incrementally, on demand, rather than all at once (unlike LDPC-Staircase repair symbols for instance). Their number can also be rather high since it is only limited by the finite field size, usually $GF(2^8)$. Said differently, GLDPC-Staircase codes can easily and dynamically be turned into small rate codes.

2.5 Decoding method

To recover erased source symbols, in addition to the (IT+RS) decoding method, we proposed a new decoding approach called hybrid (IT/RS/ML) decoding.

Let us consider a GLDPC-Staircase(N_G, K) code, built from an LDPC-Staircase(N_L, K) base code.

2.5.1 (IT+RS) decoding

The (IT+RS) decoding, for both schemes A and B, consists of a joint use of

- the IT decoder over the LDPC-Staircase graph. Extra-repair symbols are ignored at this step. This decoder features a linear complexity but also sub-optimal erasure recovery capabilities;

- the RS decoder over a given generalized check node. This is a classic RS decoding that takes into account the three types of symbols. This decoder features a higher complexity but is *MDS*;

Example 2.2. Figure 3 shows a simple example for GLDPC-Staircase code, scheme A, with $N_G = 12$, $K = 4$, $N_1 = 3$, and $r_L = \frac{1}{2}$. Here we assume that only symbols $\{S_1, P_1, P_2, P_3, e_1, e_2\}$ have been received. The receiving order for these symbols is $\{S_1, P_1, P_2, e_2, e_1, P_3\}$ (i.e., symbol transmission order is random). After receiving the first four symbols, the RS decoder triggers on the second generalized check node. This node is associated with RS(6, 4) code which recovers the (S_2, S_3) erased symbols in step 2. Then these recovered symbols trigger the SPC decoding on the first generalized check node which recovers S_4 in step 3. Decoding is successful.

Finally, Algorithm 1 details the (IT+RS) decoding that works symbol per symbol, in a recursive manner. This algorithm does not necessarily use all the received symbols: IT decoding is always preferred to RS decoding if both are possible, in order to reduce decoding complexity.

2.5.2 Hybrid (IT/RS/ML) decoding

We propose an hybrid (IT/RS/ML) decoding, generalization of the decoding approach proposed for LDPC codes in [17, 18]. Hybrid (IT/RS/ML) decoding consists of a joint use of IT, RS, and (binary/Non binary) ML decoding to achieve the performance of ML decoding at a lower complexity. It works as follows. It starts with (IT+RS) decoding. If (IT+RS) decoding succeeds, the hybrid decoding succeeds. Otherwise, the receiver switches to ML decoding, using the simplified linear system that results from the (IT+RS) decoding.

During ML decoding, we use the following decoders:

- Binary ML decoder: extra-repair symbols are ignored at this step and instead it only considers binary equations, made of simple XOR sums, in order to reduce complexity. ML decoding can consist of simple Gaussian elimination (GE) on this sub-system.
- Non-binary ML decoder: the full linear system is considered here and GE is performed on $GF(2^b)$. As in binary ML decoding, this step also features a quadratic complexity but operations are now significantly more complex (performed on $GF(2^b)$ instead of simple XOR). However, it allows reaching the maximum correction capabilities of the code.

The hybrid (IT/RS/ML) decoding algorithm is presented in Algorithm 2.

Algorithm 1 (IT+RS) decoding algorithm

```

1: for symp ← each of the  $N_b$  symbols received do
2:   Function: Decode with new symbol “symp”
3:   if all erased source symbols are recovered then
4:     ▷ No need to go any further if finished
5:     return status OK
6:   end if
7:   if  $1 \leq \text{symbol index} \leq N_L$  then ▷ LDPC symbol
8:     so try IT decoding
9:     Select IT decoding
10:    if IT decoding possible then
11:      Recover symbols ▷ recover source or
12:      LDPC repair symbols
13:    end if
14:    for New_symp ← each recovered symbol
15:      do
16:        Decode with new symbol “New_symp”
17:      end for
18:    end if
19:    Select RS decoding ▷ Continue, no matter
20:    whether symp is an LDPC or extra repair symbol
21:    if RS decoding possible then
22:      Recover symbols ▷ recover source or LDPC
23:      repair symbols
24:    end if
25:    for New_symp ← each recovered symbol do
26:      Decode with new symbol “New_symp”
27:    end for
28:  end if
29: end for

```

3 Asymptotic analysis method

3.1 Preliminaries

In the sequel, we denote by d_{vmax} and d_{cmax} respectively the maximum variable and check node degrees in the bipartite (Tanner) graph associated with LDPC-Staircase. Following [19], we define the *edge-perspective* DD polynomials by $(\lambda(x), \rho(x))$ and the *node perspective* DD polynomials by $(L(x) = \lambda(x) = \sum_{i=1}^{d_{vmax}} \lambda_i x^{i-1}, R(x) = \sum_{i=1}^{d_{cmax}} \rho_i x^{i-1})$.

Given a GLDPC-Staircase code, DD pair (λ, ρ) are defined by the underlying LDPC-Staircase code, defined by the bottom graph of Fig. 1 (that is, not containing the extra-repair nodes). Assume that transmission takes place over an erasure channel with parameter ε . We denote by $\mathcal{E}(\lambda, \rho, f_e)$ the ensemble of GLDPC-Staircase with DD pair

Algorithm 2 (IT/RS/ML) decoding algorithm

```

1: Perform (IT+RS) decoding
2: if all erased source symbols are recovered then ▷
   Check the end of decoding
3:   return status OK
4: else ▷ Trigger ML decoding
5:   Do binary ML decoding ▷ Using the simplified  $H_L$ 
   sub-system
6:   if all erased source symbols are recovered then ▷
   Check the end of decoding
7:     return status OK
8:   else
9:     Do non binary ML decoding ▷ Using the
   simplified  $H_L$  and the extra-repair equations
10:    if all erased source symbols are recovered then
    ▷ Check the end of decoding
11:      return status OK
12:    else
13:      return status Failed
14:    end if
15:  end if
16: end if

```

(λ, ρ) and with f_e the fraction of generalized check nodes with e extra-repair symbols as presented in Eq. (7).

3.2 Density evolution

3.2.1 Introduction

Over erasure channels, DE becomes one-dimensional, and it allows to analyze and even to construct capacity-achieving codes [20]. It works by recursively tracking the

erasure probability messages passed around the edges of the graph during IT decoding. Roughly speaking, this means that it recursively computes the fraction of erased messages passed during the IT decoding. Using this technique, the decoding threshold of codes is defined as the supremum value of ε (that is, the worst channel condition) that allows transmission with an arbitrary small error probability assuming N goes to infinity [19].

Let us determine the DE equations of GLDPC-Staircase codes.

3.2.2 DE equations of GLDPC-Staircase codes

Assume that an arbitrary GLDPC-Staircase code from $\mathcal{E}(\lambda, \rho, f_e)$, with length N_G goes to infinity.

We are interested in the erasure probability of messages exchanges by the (IT+RS) decoding along the messages of the LDPC-Staircase code using extra-repair variable nodes. We denote by

- P_ℓ , the probability of an LDPC symbol (source or repair) node sending an erasure at iteration ℓ to the connected generalized check nodes. Clearly, P_0 is equal to the channel erasure probability ε .
- Q_ℓ , the probability of a generalized check node sending an erasure (to an LDPC symbol-node) at iteration ℓ .

The calculus of these probabilities depends on the coding scheme used to design the GLDPC-Staircase code (scheme A or B). Next, we give more details for each case. At iteration ℓ , the LDPC symbols are erased with probability P_ℓ , while extra-repair symbols are always erased with probability ε (the channel erasure probability).

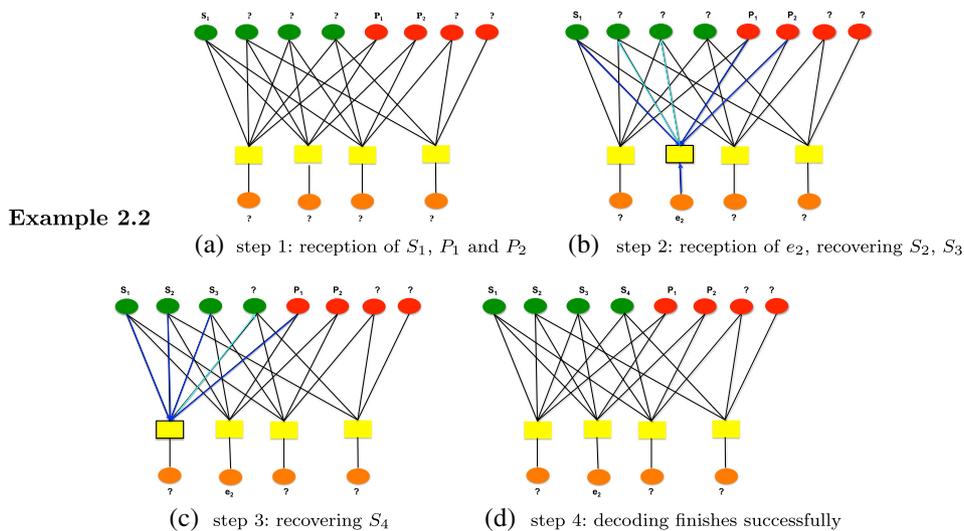


Fig. 3 (IT+RS) decoding. Figure showing an example of (IT+RS) decoding on the graph of GLDPC-Staircase code, scheme A. **a** Step 1: reception of S_1, P_1 , and P_2 . **b** Step 2: reception of e_2 , recovering S_2, S_3 . **c** Step 3: recovering S_4 . **d** Step 4: decoding finishes successfully

Scheme A: The first repair symbol generated by any RS code is one of the repair symbols of the LDPC-Staircase code.

Consider a generalized check node c connected to symbol-nodes $(v_1, \dots, v_d, e_{1,c}, \dots, e_{e(c),c})$ where v_i denotes an LDPC (source or repair) symbol node and $e_{i,c}$ denotes the i th extra-repair symbol node. Since c corresponds to an RS code, it can recover the value of an LDPC symbol node, say v_1 , if and only if the number of erasures among the other symbol-nodes $(v_2, \dots, v_d, e_{1,c}, \dots, e_{e(c),c})$ is less than or equal to $e(c)$.

It follows that the probability of a generalized check node c recovering the value of an LDPC symbol at iteration $\ell + 1$, denoted by $\bar{Q}_{\ell+1,A}(d, e(c))$, is given by:

$$\bar{Q}_{\ell+1,A}(d, e(c)) = \sum_{\substack{0 \leq i < d, 0 \leq j \leq e(c) \\ i+j \leq e(c)}} \binom{d-1}{i} P_{\ell,A}^i (1 - P_{\ell,A})^{d-1-i} \binom{e(c)}{j} \varepsilon^j (1 - \varepsilon)^{e(c)-j} \quad (11)$$

Hence, the probability of a generalized check node c sending an erasure to an LDPC symbol at iteration $\ell + 1$ is $(1 - \bar{Q}_{\ell+1,A}(d, e(c)))$. Averaging over all possible values of d and $e(c)$, we get

$$Q_{\ell+1,A} = 1 - \sum_{d=1}^{d_{cmax}} \rho_d \sum_{e=0}^E f_e \bar{Q}_{\ell+1,A}(d, e) \quad (12)$$

Scheme B: All the LDPC-Staircase repair symbols are source symbols for the RS codes.

Consider a constraint node c connected to symbol-nodes $(v_1, \dots, v_d, e_{1,c}, \dots, e_{e(c),c})$ where v_i denotes an LDPC (source or repair) symbol node and $e_{i,c}$ denotes the i th extra-repair symbol node. The node c corresponds both to a parity check constraint between LDPC symbol nodes (v_1, \dots, v_d) and to an RS linear constraint between all the symbol-nodes $(v_1, \dots, v_d, e_{1,c}, \dots, e_{e(c),c})$.

Thus, c can recover the value of an LDPC symbol node, say v_1 , if and only if one of the following (disjoint conditions) holds:

- (1) there are no erased symbols among v_2, \dots, v_d (i.e., LDPC decoding);
- (2) there is at least one erased symbol among v_2, \dots, v_d , but the number of erasures among all the symbol-nodes $(v_1, \dots, v_d, e_{1,c}, \dots, e_{e(c),c})$ is less than or equal to $e(c) - 1$.

The second condition is also equivalent to the following one:

- (2') the number of erased symbols among v_2, \dots, v_d is equal to i and the number of erased symbols among

$e_{1,c}, \dots, e_{e(c),c}$ is equal to j , with $1 \leq i \leq \min(d-1, e(c)-1)$ and $0 \leq j \leq e(c) - 1 - i$.

It follows that the probability of a generalized check node c recovering the value of an LDPC symbol at iteration $\ell + 1$, denoted by $\bar{Q}_{\ell+1,B}(d, e(c))$, is given by

$$\bar{Q}_{\ell+1,B}(d, e(c)) = (1 - P_{\ell,B})^{d-1} + \sum_{i=1}^{\min(d-1, e(c)-1)} \sum_{j=0}^{e(c)-1-i} P_{\ell,B}^i (1 - P_{\ell,B})^{d-1-i} \binom{e(c)}{j} \varepsilon^j (1 - \varepsilon)^{e(c)-j} \quad (13)$$

Averaging over all possible values of d and $e(c)$, we get

$$Q_{\ell+1,B} = 1 - \sum_{d=1}^{d_{cmax}} \rho_d \sum_{e=0}^E f_e \bar{Q}_{\ell+1,B}(d, e) \quad (14)$$

Remark 3.1. For both schemes with regular distribution of extra-repair symbols, all the generalized check nodes have E extra-repair symbols, the Eqs. (12) and (14) are reduced to:

$$Q_{\ell+1} = 1 - \sum_{d=1}^{d_{cmax}} \rho_d \bar{Q}_{\ell+1}(d) \quad (15)$$

Conversely, for both schemes, an LDPC symbol node v of degree d , connected to generalized check nodes c_1, \dots, c_d , sends an erasure to c_1 iff it was erased by the channel, and it received erased messages from all generalized check nodes c_2, \dots, c_d . Since this happens with probability $\varepsilon \cdot Q_{\ell+1}^{d-1}$, and averaging over all possible degrees d , we get

$$P_{\ell+1} = \varepsilon \sum_{d=1}^{d_{vmax}} \lambda_d Q_{\ell+1}^{d-1} = \varepsilon \lambda(Q_{\ell+1}) \quad (16)$$

For both schemes, using Eqs. (11) or (13), (12) or (14), and (16), we can determine a recursive relation between P_ℓ and $P_{\ell+1}$, with $P_0 = \varepsilon$.

The decoder can recover from a fraction of ε erased symbols iff $\lim_{\ell \rightarrow +\infty} P_\ell = 0$. This means that, when $l \rightarrow +\infty$, the (IT+RS) decoding succeeds if the DE recursion converges to zero. Then, the (IT+RS) decoding threshold of an GLDPC-Staircase code over an erasure channel is defined as the supremum value of ε such that the DE recursion converges to zero. Therefore, the (IT+RS) decoding threshold can be computed by

$$\varepsilon^{(IT+RS)}(\mathcal{E}) = \max \left\{ \varepsilon \mid \lim_{\ell \rightarrow +\infty} P_\ell = 0 \right\}. \quad (17)$$

If we transmit at $\varepsilon \leq \varepsilon^{(IT+RS)}$, then all the erased LDPC symbols can be recovered. But if we transmit at $\varepsilon > \varepsilon^{(IT+RS)}$, then some or all the erased LDPC symbols remain erased after the decoding ends.

Additionally, using the DE recursion equation, we can plot the evolution of the (IT+RS) decoding process of an GLDPC-Staircase code for an erasure channel probability ε by tracing $P_{\ell+1} = f(P_{\ell})$ with $l \rightarrow +\infty$ as shown in the following example.

Example 3.1. Let us consider a GLDPC-Staircase (scheme A) code with the following parameters:

- Rate: $r_G = \frac{1}{2}$
- Base code: $r_L = 0.8, N_1 = 5$

$$DD : \begin{cases} \lambda(x) = 0.0909.x^1 + 0.9091.x^4, \rho(x) = x^{21} \\ L(x) = 0.2.x^2 + 0.8.x^5, R(x) = x^{22} \end{cases} \quad (18)$$

- $E = 3$ (regular distribution of extra-repair symbols).

Figure 4 provides the evolution of erasure probability during the (IT+RS) decoding of GLDPC-Staircase at $\varepsilon = 0.3$. The initial fraction of erasure messages emitted by the LDPC variable nodes is $P_0 = 1$. After an iteration (at the next output of the LDPC variable nodes), this fraction has evolved to $P_1 = 0.3$. After second full iteration, i.e., at the output of the LDPC variable nodes, we see an erasure fraction of $P_2 = 0.2555$. This process continues in the same fashion for each subsequent iteration, corresponding graphically to a staircase function which is bounded above by $P_{\ell+1} = P_{\ell}$ and below by P_{out} .

3.3 EXIT functions of GLDPC-Staircase codes

3.3.1 Introduction

EXIT technique is a tool for predicting the convergence behavior of iterative processors for a variety of communication problems [21]. Over erasure channel, to visualize the convergence of iterative systems, rather than mutual information, the entropy information can be used (i.e., one minus mutual information). It is natural to use entropy in the setting of the erasure channel since the parameter ε itself represents the channel entropy. We focused in our work on EXIT based on entropy to evaluate the performance of GLDPC-Staircase codes under (IT+RS) and ML decoding. Therefore, we extended the method presented in [22]. These EXIT functions are based on DE equations derived in Section 3.2. The EXIT technique defined in this section relates to the asymptotic performance of the ensemble $\mathcal{E}(\lambda, \rho, f_e)$ under the decoding.

3.3.2 (IT+RS) EXIT function: $h^{(IT+RS)}(\varepsilon)$

The (IT+RS) EXIT function of GLDPC-Staircase code is denoted by $h^{(IT+RS)}(\varepsilon)$. It corresponds to running an (IT+RS) decoder on a very large LDPC-Staircase graph that is connected to the extra-repair variable nodes at ε until the decoder reaches a fixed point. This fixed point defines the stability of erasure probability improvement during decoding iterations. The extrinsic erasure probability of the LDPC-Staircase symbols at this fixed point gives the (IT+RS) EXIT function. Therefore, consider an $\mathcal{E}(\lambda, \rho, f_e)$, the EXIT function of the GLDPC-Staircase

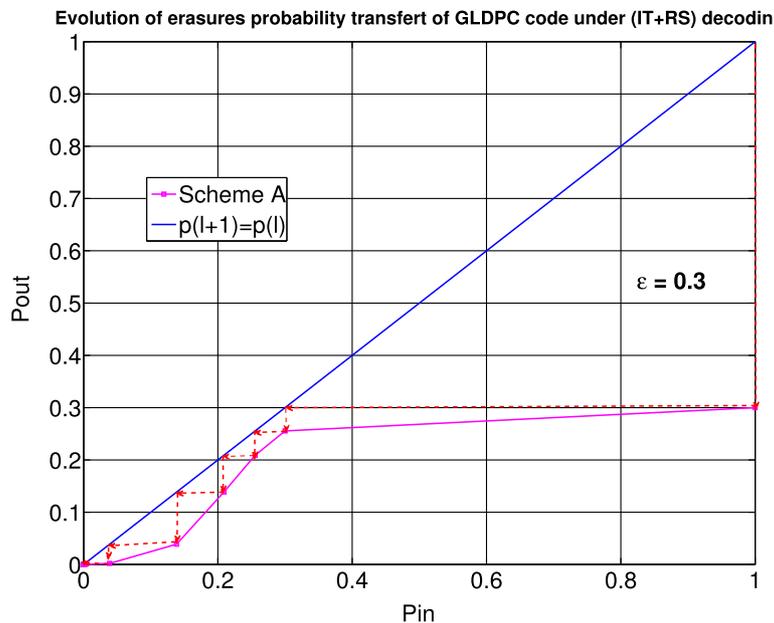


Fig. 4 DE of GLDPC-Staircase codes. Figure showing the evolution, for scheme A, of the (IT+RS) decoding process for LDPC-Staircase with $r_L = 0.8$ and $N_1 = 5$. $r_G = \frac{1}{2}, E = 3$, and $\varepsilon = 0.3$. Shannon limit = 0.5, threshold $\varepsilon^{(IT+RS)} = 0.3443$

codes under (IT+RS) decoding, over erasure channel (ϵ), is equal to the following equation:

$$h^{(IT+RS)}(\epsilon) = \frac{1}{N_L} \sum_{i=1}^{N_L} h_i^{(IT+RS)}(\epsilon) \quad (19)$$

where, $h_i^{(IT+RS)}$ is the extrinsic (IT+RS) erasure probability of LDPC-Staircase symbol “ i ” as shown in Fig. 5. $h^{(IT+RS)}(\epsilon)$ is the asymptotic (average on all the LDPC variable nodes, $N_L \rightarrow +\infty$) extrinsic erasure probability at the output of an (IT+RS) decoding. This function value can be easily computed using the DE equations of GLDPC-Staircase codes. After an infinite number of iterations of the DE recursion (Eq. (16)), the (IT+RS) decoder reaches a fixed point (i.e., $P_{\ell+1} = P_\ell, \ell \rightarrow +\infty$).

Hence, we can also write

$$h^{(IT+RS)}(\epsilon) = L(Q_{+\infty}) \quad (20)$$

where $Q_{+\infty}$ is Q_ℓ , derived from the DE equations of GLDPC-Staircase codes in Section 3.2, when the number of iterations goes to infinity.

Next, we present how can visualize the evolution of extrinsic erasure probability during (IT+RS) decoding in a graph called EXIT curve.

3.3.3 (IT+RS) EXIT curve

The (IT+RS) EXIT curve of the GLDPC-Staircase code under (IT+RS) decoding can be derived, in terms of extrinsic erasure probability (at the output of the decoder)

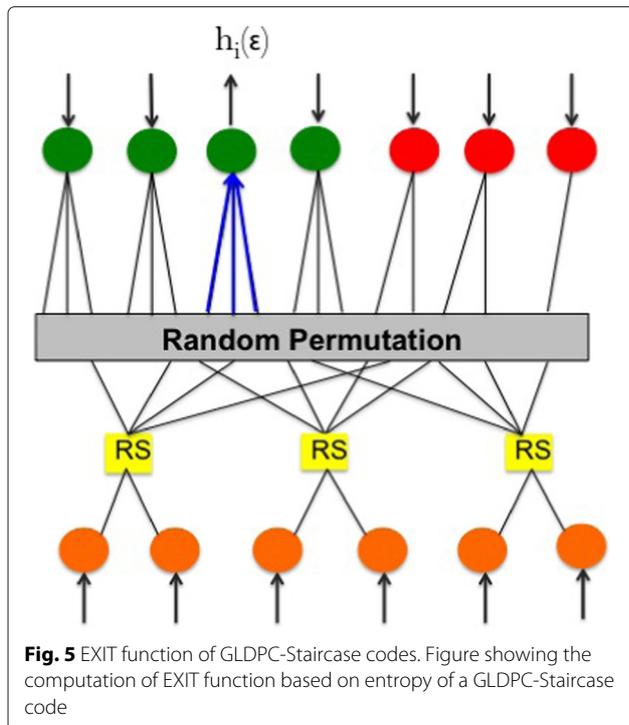


Fig. 5 EXIT function of GLDPC-Staircase codes. Figure showing the computation of EXIT function based on entropy of a GLDPC-Staircase code

as a function of the *a priori* erasure probability (input of the decoder, ϵ).

Therefore, the asymptotic (IT+RS) EXIT curve, denoted by $h^{(IT+RS)}$, is given in a parametric form by

$$h^{(IT+RS)}(\epsilon) = \begin{cases} 0 & \text{if } \epsilon \in [0, \epsilon^{(IT+RS)}] \\ L(Q_{+\infty}) & \text{if } \epsilon \in [\epsilon^{(IT+RS)}, 1] \end{cases} \quad (21)$$

Summarizing, the (IT+RS) EXIT curve is the trace of $h^{(IT+RS)}(\epsilon)$ equation for ϵ starting from $\epsilon = \epsilon^{(IT+RS)}$ until $\epsilon = 1$. In other hand, it is zero up to the (IT+RS) decoding threshold $\epsilon^{(IT+RS)}$. It then jumps to a non-zero value and also continues smoothly until it reaches one at $\epsilon = 1$. Therefore, by using this curve, $\epsilon^{(IT+RS)}$ is given by the value of ϵ where $h^{(IT+RS)}(\epsilon)$ drops down to zero.

Example 3.2. Given a GLDPC-Staircase code with rate $r_G = \frac{1}{3}$, 2 extra-repair symbols per generalized check nodes (regular distribution) and base code with the following parameters:

- $r_L = 0.6, N1 = 5$
- DD:

$$\begin{cases} \lambda(x) = 0.2105x^1 + 0.7895x^4, \rho(x) = x^9, \\ L(x) = 0.4x^2 + 0.6x^5, R(x) = x^{10} \end{cases} \quad (22)$$

The (IT+RS) EXIT function $h^{(IT+RS)}(\epsilon)$ is depicted in Fig. 6. The (IT+RS) decoding threshold, $\epsilon^{(IT+RS)}$, is given by the point where $h^{(IT+RS)}(\epsilon)$ drops down to zero. This gives $\epsilon^{(IT+RS)} = 0.5376$. It can be seen that $h^{(IT+RS)}(\epsilon) = 0$ for values $\epsilon \leq \epsilon^{(IT+RS)}$, then it jumps to a non-zero value and continues to increase until it reaches a value of 1 for $\epsilon = 1$.

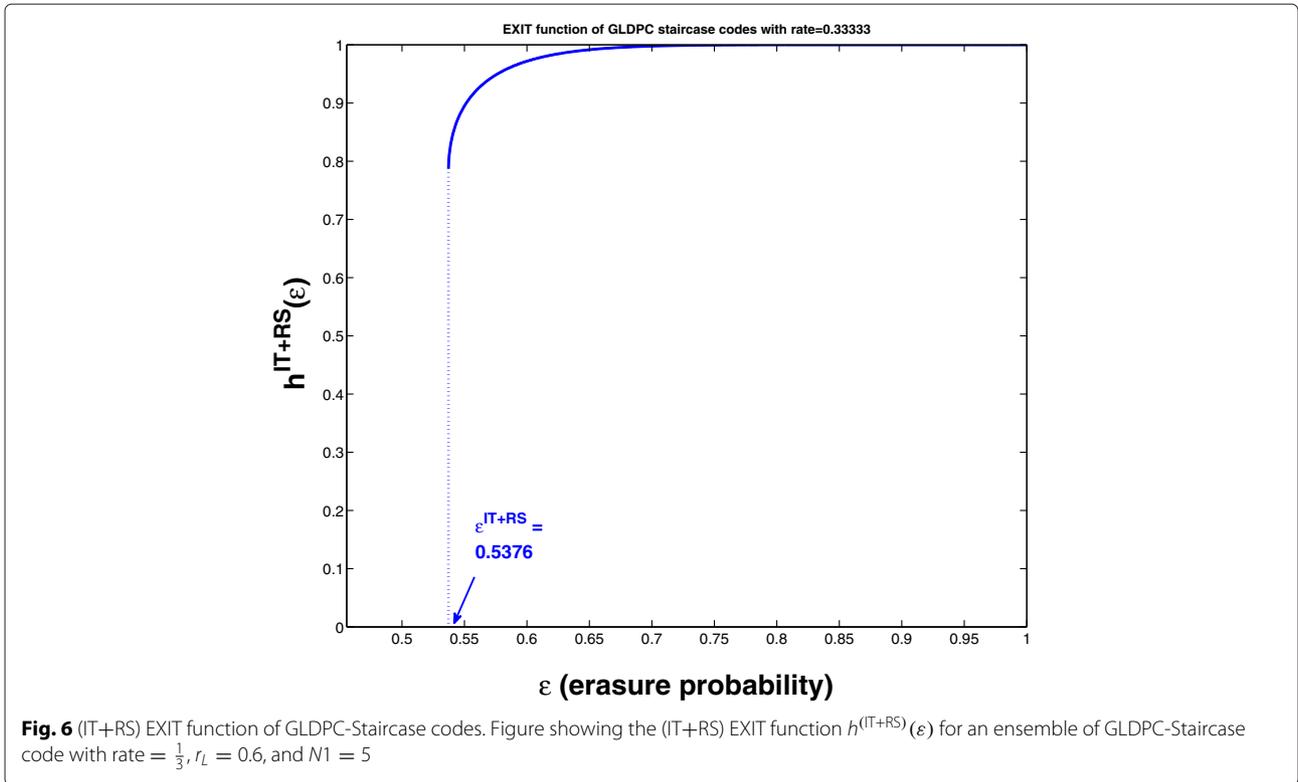
3.3.4 Upper bound on the ML decoding threshold

As for the (IT+RS) decoding, the EXIT curve of the ML decoding is also defined in terms of extrinsic erasure probability based on entropy. Precisely, in the limit of infinite code length, for a given channel erasure probability ϵ , $h^{ML}(\epsilon)$ is the probability of a symbol node being erased after ML decoding, assuming that the received value (if any) of this particular symbol has not been submitted to the decoder. The asymptotic, average on all the LDPC variable nodes, extrinsic erasure probability at the output of an ML decoding (ML EXIT function) is obtained by

$$h^{ML}(\epsilon) = \frac{1}{N_L} \sum_{i=1}^{N_L} h_i^{ML}(\epsilon) \quad (23)$$

where, $h_i^{ML}(\epsilon)$ is the extrinsic erasure probability of LDPC symbol “ i ” after ML decoding as shown in Fig. 5.

Just like LDPC codes [22], the exact computation of the EXIT function for the ML decoding is difficult. However, using the area theorem [22, 23], we have:



$$\int_{\epsilon^{ML}}^1 h^{ML}(\epsilon) = r_G, \tag{24}$$

(IT+RS) EXIT and the ML EXIT curves coincide above $\bar{\epsilon}^{ML}$.

where r_G is the designed coding rate of the given ensemble of GLDPC-Staircase codes. Moreover, since the (IT+RS) decoding is sub-optimal with respect to the ML decoding, we have $h^{(IT+RS)}(\epsilon) \geq h^{ML}(\epsilon)$. Hence, if for some $\bar{\epsilon}^{ML}$

$$\int_{\bar{\epsilon}^{ML}}^1 h^{(IT+RS)}(\epsilon) = r_G, \tag{25}$$

we necessarily have $\bar{\epsilon}^{ML} \geq \epsilon^{ML}$. This gives an upper bound on the ML threshold, which is easily computed using $h^{(IT+RS)}$.

The ML EXIT curve of the GLDPC-Staircase codes, $h^{ML}(\epsilon)$, can be constructed in the following manner:

- Step 1: Plot the (IT+RS) EXIT curve as parametrized in Eq. (21).
- Step 2: Determine the $\bar{\epsilon}^{ML}$ by integrate backwards from the right end of the curve where $\epsilon = 1$. The integration process stops at $\bar{\epsilon}^{ML}$ where it assures Eq. (25). This gives the upper bound $\bar{\epsilon}^{ML}$ of the GLDPC-Staircase codes.
- Step 3: The ML EXIT curve is now the curve which is zero at the left of the upper bound on the ML decoding threshold and equals to the (IT+RS) EXIT curve to the right of this decoding threshold (i.e., the

Remark 3.2. This upper bound is conjectured to be tight because the GLDPC-Staircase codes are based on LDPC-Staircase codes, which are binary codes and defined by quasi-regular graphs.

Example 3.3. Consider the same code of Example 3.2.

Figure 7 shows the (IT+RS) EXIT curve ($h^{(IT+RS)}(\epsilon)$) and the integral bound on ϵ^{ML} for GLDPC-Staircase code with the same distributions of Fig. 6.

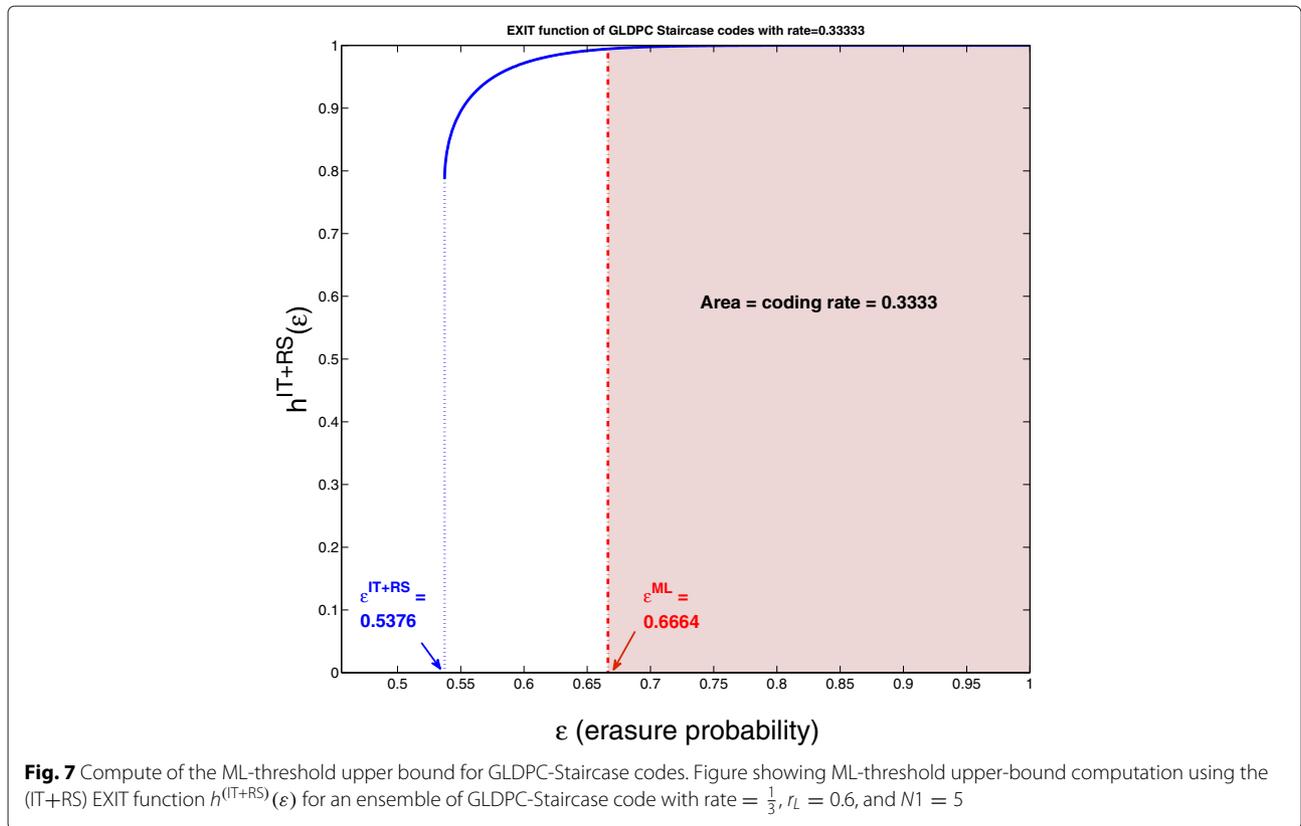
The (IT+RS) decoding threshold value is $\epsilon^{(IT+RS)} = 0.5376$.

The ML decoding threshold upper-bound is the unique point $\bar{\epsilon}^{ML} \in [\epsilon^{(IT+RS)} 1]$ such that the red area below the (IT+RS) EXIT curve, delimited by $\epsilon = \bar{\epsilon}^{ML}$ at the left and by $\epsilon = 1$ at the right, is equal to the GLDPC code rate, $r_G = 1/3$. In this case, we obtain $\bar{\epsilon}^{ML} = 0.6664$.

4 Optimization of GLDPC-Staircase codes

4.1 Description

GLDPC-Staircase codes can be viewed as an extension of LDPC-Staircase code (base code) into generalized LDPC-Staircase code using RS codes. Moreover, GLDPC-Staircase codes can be constructed using two structures which differ in the type of the generated LDPC repair



symbols that are either RS repair symbols or not, as follows:

- Scheme A has the property that on each generalized check node, the repair symbol generated by the LDPC code is also an RS repair symbol.
- On the opposite, with scheme B the generated LDPC repair symbol, on each generalized check node, is an RS source symbol.

In addition, the configuration of GLDPC-Staircase codes depends on the important internal parameters, namely

- the extra-repair symbols distribution across the H_L rows: regular distribution or irregular uniform distribution,
- the $N1$ parameter of the base code: degree of source variable nodes in H_L ,
- the base code rate r_L .

Therefore, in this section, we start by showing the impacts of the property that the generated LDPC repair symbols are at the same time RS repair symbols, on the decoding behavior (i.e., compare scheme A and scheme B). Then, the best configuration of these parameters for hybrid (IT/RS/ML) decoding³ will be discussed. To gauge the correction capabilities of decoding, we use

the asymptotic analysis based on DE and EXIT techniques presented in Section 3, as well as the finite length analysis.

4.2 Experimental conditions

For the finite length analysis, we have developed a GLDPC-Staircase codec based on RS codes under (IT+RS) and ML decoding methods, in C language, using the OpenFEC.org project (<http://openfec.org>). All experiments are carried out by considering a memory-less erasure channel along with a transmission scheme where all the source and repair symbols are sent in a fully random order.

This has the benefit to make the performance results independent of the loss model⁴ and the target channel loss rate is the only parameter that needs to be considered.

Different LDPC-Staircase matrices are used (more precisely we change the PRNG seed used to create the matrix). Then, the results, averaged over the tests obtained by varying LDPC-Staircase matrix, show the average behavior of GLDPC-Staircase codes.

In the sequel, we evaluate the finite length performance based on the decoding overhead⁵, the decoding inefficiency ratio⁶ and the failure decoding probability⁷.

For the asymptotic analysis, we use commonly the following DD of LDPC-Staircase codes as presented in Tables 1 and 2.

The calculus of these degree distributions is based on the parameter $N1$, $d_{r_{H1}}$, and the structure of LDPC-Staircase codes.

For an irregular uniform distribution of extra-repair symbols, we use the notation $f(\%) = [f_0 f_1 f_2 \dots f_e]$ to define the fractions of generalized check nodes with e extra-repair symbols. For example, $f(\%) = [25 \ 50 \ 25]$ means that we have 25 % of generalized check nodes have 0 extra-repair symbols, 50 % of generalized check nodes have 1 extra-repair symbols, and 25 % of generalized check nodes have 2 extra-repair symbols.

4.3 Best coding scheme for GLDPC-Staircase codes

Throughout this section, we investigate the impacts of the property given by scheme A on decoding performance in different configurations of GLDPC-Staircase codes. For this reason, the study allows to determine the best for the hybrid (IT/RS/ML) decoding through a comparison between scheme A and scheme B.

4.3.1 Asymptotic results

Let us consider a base code with distribution defined in Table 1 for $r_L = 0.8$ and $N1 = 5$. We use the DE equations proposed in Section 3.2 to plot in Fig. 8 the evolution of the erasure probability transfer on the graph of GLDPC-Staircase code with $r_G = \frac{1}{2}$ and $E = 3$ (regular distribution) for scheme A ($P_{\ell+1,A} = f(P_{\ell,A})$) and scheme B ($P_{\ell+1,B} = f(P_{\ell,B})$). These curves represent the value of the erasure probability on all the LDPC symbols during the propagation of the erasure probability between generalized check nodes and variable nodes of the GLDPC-Staircase tanner graph where (ε) equals to 0.32.

This figure shows that the initial fraction of erasure messages emitted by the LDPC variable nodes is $P_l = 1$ in schemes A and B. After an iteration (at the next output of the LDPC variable nodes) this fraction has evolved to $P_{l+1} = 0.32$ for the two schemes.

After second full iteration, i.e., at the output of the LDPC variable nodes, we see that an erasure fraction of scheme A is equal to $P = 0.2889$ whereas it is equal to $P = 0.3117$ for scheme B. This difference explains

that the erasure probability in scheme A decreases more quickly than scheme B (i.e., the correction of the erasure in scheme A is better than scheme B). After that the process of the transfer continues in the same fashion for each subsequent iteration.

The figure also shows that the process of DE for scheme B is stuck at value >0 ($P = 0.3094$) while for scheme A the process finishes with $P = 0$. This means that under (IT+RS) decoding and at $\varepsilon = 0.32$, the GLDPC-Staircase codes converge (i.e., can recover all the erased LDPC symbols) only with scheme A. We continue the comparison between the two schemes in terms of *decoding threshold* using the EXIT analysis presented in Section 3.3. This analysis allows us to compute the (IT+RS) decoding threshold ($\varepsilon^{(IT+RS)}$) and the upper bound on the ML decoding threshold ($\bar{\varepsilon}^{ML}$). We note that DE also allows to determine the decoding threshold, but it requires several calculations.

Table 3 provides the comparison in terms of $\varepsilon^{(IT+RS)}$ and $\bar{\varepsilon}^{ML}$ between scheme A and scheme B (with regular distribution and $r_L = \frac{2}{3}$) for two global code rates ($\frac{1}{2}$ and $\frac{2}{5}$) and different values of $N1$. This table reveals that for different values of $N1$, scheme A outperforms scheme B under (IT+RS) decoding. Therefore, the property that the generated LDPC repair symbols are RS repair symbols helps to get closer to channel capacity limit. Whereas under ML decoding, below of $N1 = 5$, scheme B is preferable and beyond this value, this property has no great significant impact. This is explained as follows. In practice, the efficiency of the ML decoder over BEC is related to the densification of its linear system. Therefore, low value of $N1$ implies a sparse binary linear system of LDPC-Staircase codes which causes degradation on the ML decoding results [24]. Whereas, as mentioned in Section 2.5.2, the linear system of GLDPC-Staircase codes is composed of a binary sub-system (composed from LDPC-Staircase equations) and a non binary sub-system (composed from extra-repair equations) which is somewhat more dense with scheme B than scheme A. Therefore, this difference has an impact on performance of global system when the binary sub-system is sparse; otherwise, it is vanished. We will see next, that $N1 = 5$ is the best value for the hybrid decoding type, therefore we prefer scheme A in this case.

Table 1 Used degree distribution. Table showing LDPC-Staircase DD for different values of $N1$ where $r_L = \frac{2}{3}$

$N1$	edge DD ($\lambda(x), \rho(x)$)	node DD ($L(x), R(x)$)
3	$\lambda(x) = 0.25x^1 + 0.75x^2, \rho(x) = x^7$	$L(x) = 0.3333x^2 + 0.6667x^3, R(x) = x^8$
4	$\lambda(x) = 0.2x^1 + 0.8x^3, \rho(x) = x^9$	$L(x) = 0.3333x^2 + 0.6667x^4, R(x) = x^{10}$
5	$\lambda(x) = 0.1666x^1 + 0.8333x^4, \rho(x) = x^{11}$	$L(x) = 0.3333x^2 + 0.6667x^5, R(x) = x^{12}$
6	$\lambda(x) = 0.1429x^1 + 0.8571x^5, \rho(x) = x^{13}$	$L(x) = 0.3333x^2 + 0.6667x^6, R(x) = x^{14}$

Table 2 Used degree distribution. Table showing LDPC-Staircase DD for different values of r_L where $N1 = 5$

r_L	edge DD ($\lambda(x), \rho(x)$)	node DD ($L(x), R(x)$)
$\frac{1}{2}$	$\lambda(x) = 0.2857x^1 + 0.7143x^4, \rho(x) = x^6$	$L(x) = 0.5000x^2 + 0.5000x^5, R(x) = x^7$
$\frac{2}{3}$	$\lambda(x) = 0.1666x^1 + 0.8333x^4, \rho(x) = x^{11}$	$L(x) = 0.3333x^2 + 0.6667x^5, R(x) = x^{12}$
0.75	$\lambda(x) = 0.1176x^1 + 0.8824x^4, \rho(x) = x^{16}$	$L(x) = 0.2500x^2 + 0.75x^5, R(x) = x^{17}$
0.8	$\lambda(x) = 0.0909x^1 + 0.9091x^4, \rho(x) = x^{21}$	$L(x) = 0.2x^2 + 0.8x^5, R(x) = x^{22}$

In the previous analysis, we fixed the base code rate and the distribution type of extra-repair symbols to study only the impact of the property of scheme A when varying $N1$. Let us now see the impact of this property when we vary r_L (i.e., vary E) and the distribution of extra-repair symbols with $N1$ fixed to 5.

Table 4 provides the comparison in terms of $\varepsilon^{(IT+RS)}$ and $\bar{\varepsilon}^{ML}$ between scheme A and scheme B (with $r_G = \frac{1}{2}$) for different values of r_L (i.e., vary E) using a regular distribution of extra-repair symbols. This table proves that, for different values of $E > 0$, the (IT+RS) decoding threshold of scheme A is higher than that of scheme B. On the opposite, the ML decoding thresholds of the two schemes, for different values of E , are almost equivalent.

Additionally, for an irregular uniform distribution of extra-repair symbols, Table 5 also shows that, for different values of f (distribution of extra-repair symbols), scheme A is better than scheme B for (IT+RS) decoding

and both achieve the same ML decoding thresholds.

Let us move to see the results when varying the rate of GLDPC-Staircase code. Table 6 provides the comparison in terms of $\varepsilon^{(IT+RS)}$ and $\bar{\varepsilon}^{ML}$ between scheme A and scheme B for different values of r_G with $r_L = \frac{2}{3}$, $N1 = 5$ and regular distribution. This table reveals that for different rates of GLDPC-Staircase code, scheme A outperforms scheme B under (IT+RS) decoding and both have the same behavior under ML decoding. For an irregular uniform distribution, in Table 7, we provide a comparison between decoding thresholds of the two schemes with $r_L = \frac{2}{3}$ for different values of f (distribution of extra-repair symbols). This table also shows the same results for regular distribution.

Therefore, for all configurations of GLDPC-Staircase codes, the structure of scheme A resists to the channel loss more than scheme B under (IT+RS) and both have the same behavior under ML decoding with dense system.

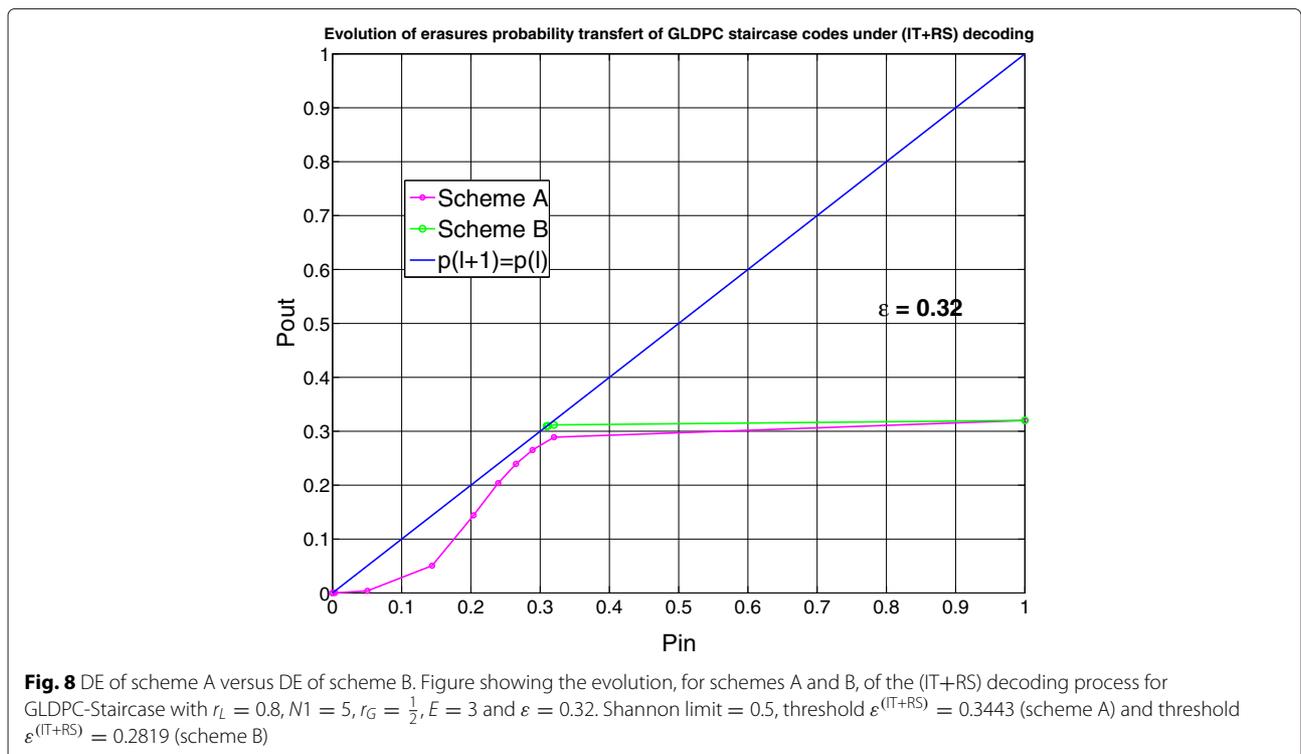


Table 3 Decoding threshold comparison between schemes A and B for different values of $N1$ for (IT+RS) decoding and ML decoding. GLDPC-Staircase codes with $r_L = \frac{2}{3}$ and regular distribution of extra-repair symbols

r_G	Decoding threshold	$N1$	Scheme A	Scheme B		
$\frac{1}{2}$ ($E = 1$)	$\epsilon^{(IT+RS)}$	3	0.4571	0.2905		
		4	0.4269	0.2867		
		5	0.3943	0.2709		
		6	0.3646	0.2536		
		$\bar{\epsilon}^{ML}$	3	0.4900	0.4985	
			4	0.4976	0.4996	
	5		0.4993	0.4997		
	6		0.4998	0.4999		
	$\frac{2}{5}$ ($E = 2$)		$\epsilon^{(IT+RS)}$	3	0.5532	0.4041
				4	0.5136	0.3878
		5		0.4744	0.3639	
		6		0.4392	0.3401	
$\bar{\epsilon}^{ML}$		3		0.5946	0.5994	
		4		0.5990	0.5998	
	5	0.5998	0.5999			
	6	0.5999	0.5999			

4.3.2 Finite length results

This section aims to give additional claims on the impact of the property of scheme A in terms of decoding inefficiency ratio, decoding overhead, decoding failure probability, and error floor. All results are determined using $N1 = 5$ and a regular distribution of extra-repair symbols.

Figures 9b and 10b provide the average (over 1000 different codes) decoding inefficiency ratio of both schemes under ML decoding for two different code rates ($\frac{1}{2}$ and $\frac{1}{3}$). They show that no matter the dimension, K , both schemes perform the same, with results quite close

Table 4 Decoding threshold comparison between (IT+RS) decoding and ML decoding, for $r_G = \frac{1}{2}$ (Shannon limit = 0.5) and regular distribution of extra-repair symbols

Decoding threshold	E	r_L	Scheme A	Scheme B
$\epsilon^{(IT+RS)}$	0	$\frac{1}{2}$	0.4380	0.4380
	1	$\frac{2}{3}$	0.3943	0.2709
	2	0.75	0.3647	0.2773
	3	0.8	0.3443	0.2819
$\bar{\epsilon}^{ML}$	0	$\frac{1}{2}$	0.4946	0.4946
	1	$\frac{2}{3}$	0.4993	0.4997
	2	0.75	0.4999	0.4999
	3	0.8	0.4999	0.4999

Table 5 Decoding threshold comparison between (IT+RS) decoding and ML decoding, for $r_G = \frac{1}{2}$ (Shannon limit = 0.5) and irregular uniform distribution of extra-repair symbols

Decoding threshold	$f(\%)$	r_L	Scheme A	Scheme B
$\epsilon^{(IT+RS)}$	[0]	$\frac{1}{2}$	0.4380	0.4380
	[25 50 25]	$\frac{2}{3}$	0.4064	0.3029
	[0 50 25 0 25]	0.75	0.3874	0.3089
	[5 10 15 20 50]	0.8	0.3589	0.3007
$\bar{\epsilon}^{ML}$	[0]	$\frac{1}{2}$	0.4946	0.4946
	[25 50 25]	$\frac{2}{3}$	0.4993	0.4997
	[0 50 25 0 25]	0.75	0.4998	0.4998
	[5 10 15 20 50]	0.8	0.4998	0.4998

to that of MDS codes (characterized by an decoding inefficiency ratio always equals to 1). This means that for small and large object size, the property of scheme A has no impact on the ML decoding inefficiency ratio. These results hold for the two considered code rates.

Figures 9a and 10a do the same in case of (IT+RS) decoding only. They show that scheme A exhibits the lowest average decoding inefficiency ratio in all cases. This is made possible by a higher number of RS repair symbols (i.e., increase of the minimum distance) for scheme A, which mechanically increases the success probability of decoding an erased symbol on each generalized check nodes. The increase of the RS repair symbols also avoids stopping sets associated to short cycles that stuck (IT+RS) decoding. This means that scheme A is more efficient on (IT+RS) decoding than scheme B.

In order to go further to see the error floor and overhead achieved by each scheme under ML decoding, we analyze the ML decoding failure probability. In Fig. 11, we plot the ML decoding failure probability versus channel loss percentage (in Fig. 11a) and versus number of received

Table 6 Decoding threshold comparison between (IT+RS) decoding and ML decoding for different values of r_G with $r_L = \frac{2}{3}$ and regular distribution of extra-repair symbols

Decoding threshold	r_G	E	Scheme A	Scheme B	Shannon limit
$\epsilon^{(IT+RS)}$	$\frac{2}{3}$	$E = 0$	0.2709	0.2709	0.3333
	$\frac{1}{2}$	$E = 1$	0.3943	0.2709	0.5
	$\frac{2}{5}$	$E = 2$	0.4744	0.3639	0.6
	$\frac{1}{3}$	$E = 3$	0.5332	0.4394	0.6667
$\bar{\epsilon}^{ML}$	$\frac{2}{3}$	$E = 0$	0.3301	0.3301	0.3333
	$\frac{1}{2}$	$E = 1$	0.4993	0.4997	0.5
	$\frac{2}{5}$	$E = 2$	0.5998	0.5999	0.6
	$\frac{1}{3}$	$E = 3$	0.6666	0.6666	0.6667

Table 7 Decoding threshold comparison between (IT+RS) decoding and ML decoding for different values of r_G with $r_L = \frac{2}{3}$ and irregular uniform distribution of extra-repair symbols

Decoding threshold	r_G	$f(\%)$	Scheme A	Scheme B	Shannon limit
$\bar{\varepsilon}^{(IT+RS)}$	$\frac{2}{3}$	[0]	0.2709	0.2709	0.3333
	$\frac{1}{2}$	[25 50 25]	0.4064	0.3029	0.5
	$\frac{2}{5}$	[0 25 50 25]	0.4819	0.3765	0.6
	$\frac{1}{3}$	[0 25 25 10 15 15 10]	0.5598	0.4768	0.6667
$\bar{\varepsilon}^{ML}$	$\frac{2}{3}$	[0]	0.3301	0.3301	0.3333
	$\frac{1}{2}$	[25 50 25]	0.4993	0.4997	0.5
	$\frac{2}{5}$	[0 25 50 25]	0.5997	0.5999	0.6
	$\frac{1}{3}$	[0 25 25 10 15 15 10]	0.6665	0.6666	0.6667

symbols (in Fig. 11b). To that purpose, we choose $r_G = \frac{1}{2}$ ($E = 1$), $K = 1000$, and 10^6 tested codes.

In Fig. 11a, the black vertical line corresponds to ideal, MDS code, for which the decoding failure is equal to 0 as long as the experienced loss rate is strictly inferior to 50 % for $r_G = \frac{1}{2}$. This figure confirms that for two schemes, the GLDPC-Staircase codes have a very small decoding failure probability, with no visible error floor above 10^{-5} . The little difference between the two curves is readable at the foot where we test several codes (i.e., at 49.45 % scheme A has $4.16 \cdot 10^{-6}$ as decoding failure probability whereas scheme B has $5.45 \cdot 10^{-6}$). Figure 11b gives additional details of the behavior of the two schemes using ML decoding. This figure confirms that scheme B has almost same decoding overhead as scheme A (i.e., with 6 symbols added to K , scheme A has $6.93 \cdot 10^{-6}$ decoding failure probability while $7.27 \cdot 10^{-6}$ with scheme B for channel erasure probability equals to 49.6 %). Also, the two schemes achieve $\simeq 5 \cdot 10^{-2}$ decoding failure probability with overhead equals to 2. Therefore, both schemes have a very small decoding overhead, close to that of MDS codes.

4.3.3 Conclusion of the analysis

The asymptotic analysis and finite length analysis confirm that all results prove that scheme A is globally the best solution: it significantly performs better than scheme B with an (IT+RS) decoding and leads similar performance to scheme B with an ML decoding with dense system. Thus, to design GLDPC-Staircase codes, with hybrid (IT/RS/ML) decoding, we must choose scheme A. Therefore, the rest of this document will only consider scheme A.

4.4 Tuning internal parameters of GLDPC-Staircase codes

In this section, we analyze the impact of three configuration parameters of GLDPC-Staircase codes on the erasure recovery performance in order to obtain the best configuration over hybrid (IT/RS/ML) decoding.

4.4.1 The extra-repair symbol distribution

As shown in Section 2.1, we can distribute the extra-repair symbols on the generalized check nodes in two ways: regular, or irregular uniform distribution. In [13], based on asymptotic results, it is shown that these codes with (IT+RS) decoding perform the best under an irregular uniform distribution rule. However, in our work, we consider also the ML decoding scheme and the situation is completely different. Therefore, we test these two distributions to determine the best on each decoding type.

Figure 12 provides the average decoding inefficiency ratio (i.e., average of 1000 GLDPC codes with $r_G = \frac{1}{2}$) of GLDPC-Staircase codes, for different object sizes. It shows that the regular distribution performs significantly better under ML decoding, both with small and large objects.

Based on asymptotic results, we give the gaps to capacity of GLDPC-Staircase codes with irregular uniform distribution and regular distribution of extra-repair symbols for different code rates r_G with $N1 = 5$ and $r_L = \frac{2}{3}$ in Table 8. In addition, we provide in Table 9 the gaps to capacity of the two distributions for different values of r_L with fixed global code rate $r_G = \frac{1}{2}$ and $N1 = 5$.

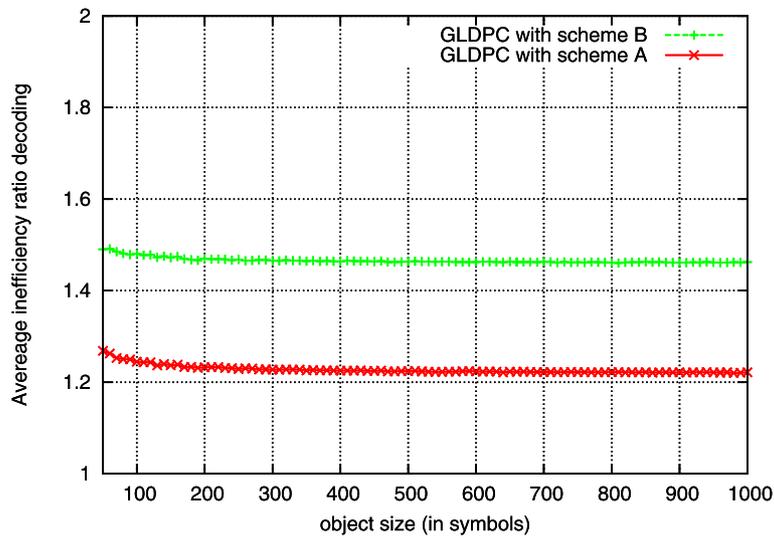
The gap to capacity (Δ) is computed using the following equation:

$$\Delta = 1 - r_G - \varepsilon_{th}, \quad (26)$$

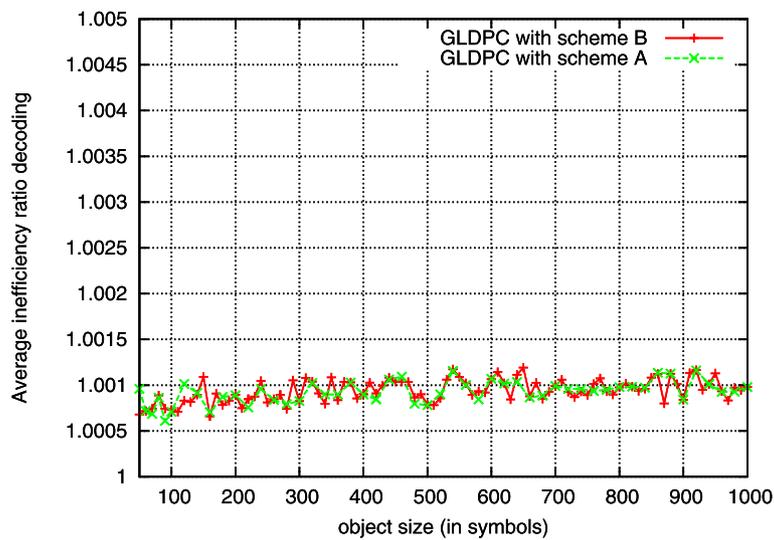
(with ε_{th} is the decoding threshold).

These tables show that, for different values of global code rate and base code rate, under (IT+RS) decoding, the GLDPC-Staircase codes produce higher gap to capacity with a regular distribution rather than with an irregular uniform distribution. While, under ML decoding, the regular distribution allows to have GLDPC-Staircase codes very close to the channel capacity more than the irregular uniform distribution. This confirms the finite length analysis.

Let us see the case where $N1$ varies. Table 10 provides the decoding thresholds of the two decoding types for



(a) (IT+RS) decoding, $r_G = \frac{1}{2}$ ($E = 1$)



(b) ML decoding, $r_G = \frac{1}{2}$ ($E = 1$)

Fig. 9 Inefficiency ratio (IT+RS) and ML decodings. Figure showing the average performance under (IT+RS) decoding **(a)** and ML decoding **(b)**, with rate $\frac{1}{2}$, as a function of K . **a** (IT+RS) decoding, $r_G = \frac{1}{2}$ ($E = 1$). **b** ML decoding, $r_G = \frac{1}{2}$ ($E = 1$)

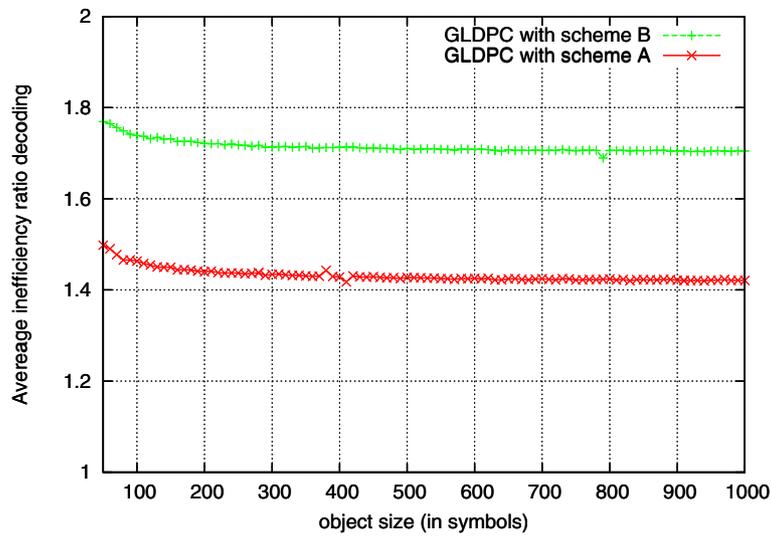
$r_L = \frac{2}{3}$ for the two distributions into different values of global code rates. This table also shows that ML decoding favors the regular distribution for different values of $N1$ (more advantage with low value of $N1$). Therefore, for different values of $N1$, r_L , and r_G , the regular distribution is accorded to the ML decoding whereas irregular uniform distribution is suitable for (IT+RS) decoding.

As our objective is to focus on the hybrid decoding and ML performance, in the remaining of this work we only

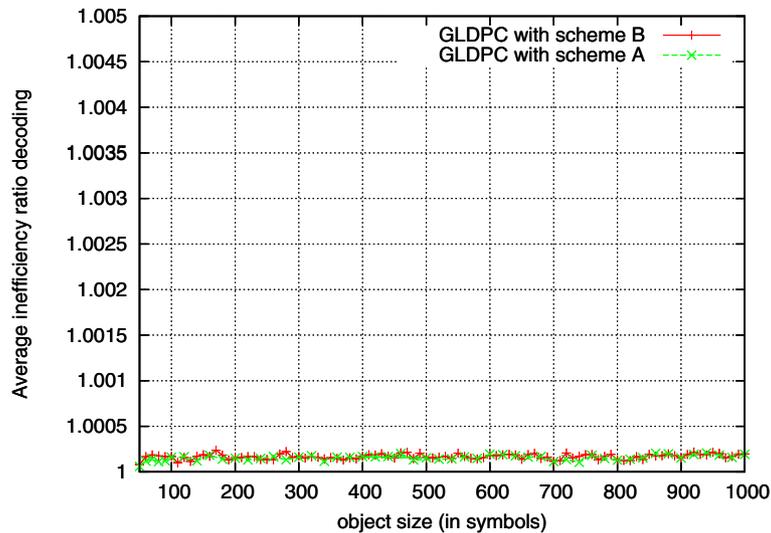
focus on the regular distribution where there are exactly $e(c) = E$ extra-repair symbols per generalized check node c .

4.4.2 $N1$ parameter

Let us now adjust the second internal parameter of the GLDPC-Staircase codes, $N1$, which represents the degree of the source variable nodes of the base code matrix H_L .



(a) (IT+RS) decoding, $r_G = \frac{1}{3}$ ($E = 3$)



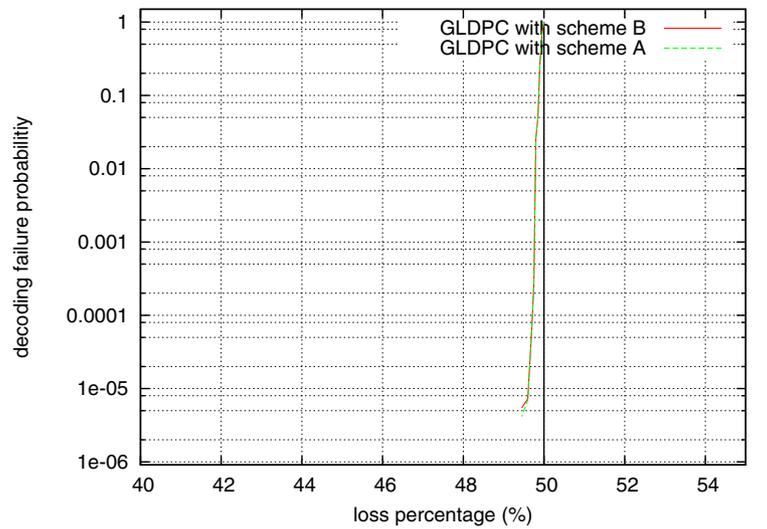
(b) ML decoding, $r_G = \frac{1}{3}$ ($E = 3$)

Fig. 10 Inefficiency ratio (IT+RS) and ML decodings. Figure showing the average performance under (IT+RS) decoding **(a)** and ML decoding **(b)**, with rate $\frac{1}{3}$, as a function of K . **a** (IT+RS) decoding, $r_G = \frac{1}{3}$ ($E = 3$). **b** ML decoding, $r_G = \frac{1}{3}$ ($E = 3$)

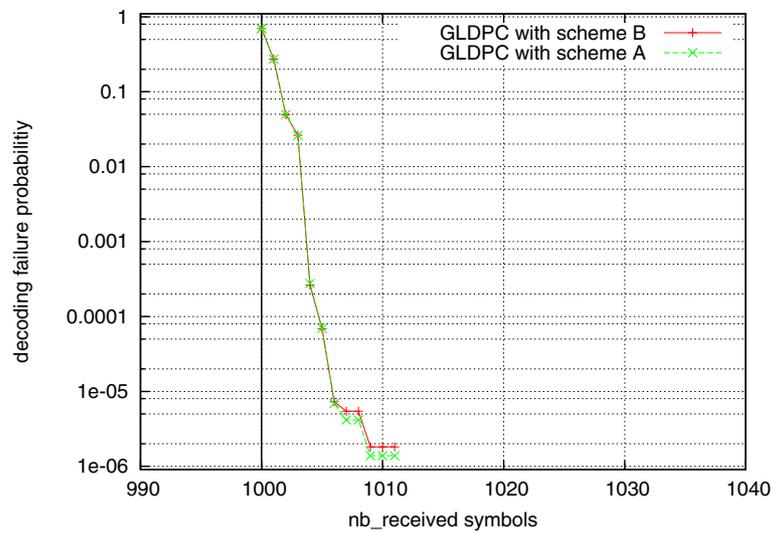
The increase of $N1$ parameter causes the “densification” of H_L and maybe the decrease of the smallest stopping set size. It is well known that the effectiveness of IT decoding over erasure channel is related to the sparseness of the LDPC graph. In addition, the correction capabilities of LDPC codes are limited by the size of smallest stopping sets.

Since, in our case we have IT and RS decoding working together, let us see if the RS decoding helps

IT decoding to prevent the negative impact of the densification of the graph on the decoding (i.e., to see if the densification also causes the degradation of (IT+RS) decoding performance). Therefore, in Table 11, we provide the average decoding inefficiency ratio (i.e., average of 1000 GLDPC-Staircase codes for each r_G) for different r_G versus $N1$ under (IT+RS) decoding with $K = 1000$ and $r_L = \frac{2}{3}$. This table shows that, for different r_G and under (IT+RS) decoding, increasing $N1$



(a)



(b)

Fig. 11 Decoding failure probability under ML decoding. Figure showing the decoding failure probability, with rate $\frac{1}{2}$ and $K = 1000$, versus the channel loss percentage in (a) and as a function of the number of received symbols in (b)

induces an increase in the average decoding inefficiency ratio. This means that the extra-repair symbols which are used to cope with the problem of small stopping sets in the base code graph do not succeed. Moreover, increasing E (i.e., reducing the GLDPC code rate) does not solve the problem. Therefore, the increase of N_1 leads to the deterioration of the ability to correct with (IT+RS) decoding. This table also shows that, for different GLDPC code rates, the behavior of ML decoding is totally different than the one observed for (IT+RS) decoding.

We give the same remarks using Table 12 where we compute the threshold of (IT+RS) decoding and ML decoding for a different value of N_1 .

Additionally, the decoding complexity depends on the number of XOR operations on the graph of IT decoding, whereas it depends on the size and the density of the linear system to be solved of ML decoding [24]. Then, the increase of N_1 has a negative impact on the IT and ML decoding complexity.

Therefore, with ML decoding, with respect to the low decoding complexity, the most significant performance

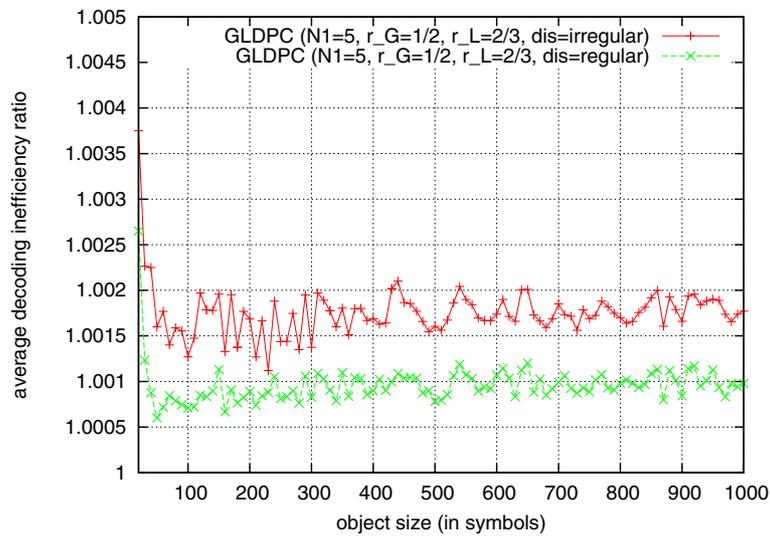


Fig. 12 Irregular uniform distribution versus regular distribution. Figure showing performance comparison between irregular uniform and regular distributions of extra-repair symbols, with $N1 = 5$, $r_L = \frac{2}{3}$, and $r_G = \frac{1}{2}$

gains are obtained by switching from $N1 = 3$ to 5. Above this value, the performance only improves slightly. This value $N1=5$ also limits the performance degradation of (IT+RS) decoding compared to values $N1 > 5$. $N1 = 5$ is the best value that can be used by the hybrid (IT/RS/ML) decoding.

4.4.3 The base code rate r_L

Let us consider a GLDPC-Staircase code rate r_G . Several values of the base code rate r_L , or equivalently of E , enable

to achieve the global code rate r_G (see Eq. (8)). However, choosing a value impacts the achieved performance.

In Fig. 13, we plot the average ML decoding inefficiency ratio (i.e., average of 1000 GLDPC codes with $r_G = \frac{1}{3}$) for a different object size. This figure shows that increasing r_L rate (i.e., increasing the number E), the average decoding inefficiency ratio quickly approaches 1 (i.e., ideal code) as $E = 3$ (i.e., $r_L = \frac{2}{3}$), even for very small code dimensions. We also apply the techniques developed in Section 3 to adjust E , by computing the upper bound on the ML decoding threshold for several values of E . These results are summarized in Table 13 and compared to the Shannon capacity limit (δ_{sh}). We notice that increasing E (or equivalently increasing the LDPC code rate) quickly increases the upper bound on the ML decoding threshold, until it reaches a stable value very close to the Shannon limit δ_{sh} . Depending on r_G , this stable value is obtained with $E = 1$,

Table 8 (IT+RS) decoding and ML decoding gaps to capacity for irregular uniform distribution (Δ_2) versus regular distribution (Δ_1) of extra-repair symbols for GLDPC-Staircase codes with $N1 = 5$ and $r_L = \frac{2}{3}$

Decoding type	r_G	E	Δ_1	$f(\%)$	Δ_2
(IT+RS)	$\frac{2}{3}$	$E = 0$	0.06243	[0]	0.06243
	$\frac{1}{2}$	$E = 1$	0.1057	[25 50 25]	0.0936
	$\frac{2}{5}$	$E = 2$	0.1256	[0 25 50 25]	0.1181
	$\frac{1}{3}$	$E = 3$	0.13346	[0 25 25 10 15 15 10]	0.1068
	$\frac{1}{3.5}$	$E = 4$	0.1350	[0 25 0 0 0 75]	0.1167
	$\frac{1}{5}$	$E = 7$	0.1268	[0 0 0 0 0 10 80 10]	0.1262
ML	$\frac{2}{3}$	$E = 0$	0.00323	[0]	0.00323
	$\frac{1}{2}$	$E = 1$	0.0007	[25 50 25]	0.0007
	$\frac{2}{5}$	$E = 2$	0.0002	[0 25 50 25]	0.0003
	$\frac{1}{3}$	$E = 3$	0.0001	[0 25 25 10 15 15 10]	0.0002
	$\frac{1}{3.5}$	$E = 4$	0.000081	[0 25 0 0 0 75]	0.0000857
	$\frac{1}{5}$	$E = 7$	0.00007	[0 0 0 0 0 10 80 10]	0.00008

Table 9 (IT+RS) decoding and ML decoding gaps to capacity for irregular uniform distribution (Δ_2) versus regular distribution (Δ_1) of extra-repair symbols for GLDPC-Staircase codes with $N1 = 5$ and $r_G = \frac{1}{2}$

Decoding type	r_L	E	Δ_1	$f(\%)$	Δ_2
(IT+RS)	$\frac{1}{2}$	0	0.062	[0]	0.062
	$\frac{2}{3}$	1	0.1057	[25 50 25]	0.0936
	0.75	2	0.1353	[0 50 25 0 25]	0.1126
ML	0.8	3	0.1557	[5 10 15 20 50]	0.1411
	$\frac{1}{2}$	0	0.0054	[0]	0.0054
	$\frac{2}{3}$	1	0.0007	[25 50 25]	0.0007
ML	0.75	2	0.0001	[0 50 25 0 25]	0.0002
	0.8	3	0.0001	[5 10 15 20 50]	0.0002

Table 10 Threshold decoding comparison between the two distributions of extra-repair symbols for different values of $N1$ and r_G with $r_L = \frac{2}{3}$

Distribution type	Decoding threshold	$N1$	$r_G = \frac{1}{2}$	$r_G = \frac{2}{5}$	$r_G = \frac{1}{3}$		
Regular distribution	$\varepsilon^{(IT+RS)}$	3	0.4571	0.5532	0.6184		
		4	0.4269	0.5136	0.5753		
		5	0.3943	0.4744	0.5332		
		6	0.3646	0.4392	0.4955		
		$\bar{\varepsilon}^{ML}$	3	0.49	0.5946	0.6633	
			4	0.4976	0.5990	0.6661	
	5		0.4993	0.5998	0.6666		
	6		0.4998	0.5999	0.6666		
	Irregular uniform distribution		$\varepsilon^{(IT+RS)}$	3	0.4605	0.5563	0.6287
				4	0.4365	0.5196	0.5959
		5		0.4064	0.4819	0.5598	
		6		0.3779	0.4476	0.5257	
$\bar{\varepsilon}^{ML}$		3		0.4863	0.5935	0.6600	
		4		0.4966	0.5987	0.6654	
	5	0.4993	0.5997	0.6666			
	6	0.4997	0.5998	0.6666			

2, or 3. Therefore, a small number of extra-repair symbols per generalized check node is sufficient to get extremely close to the channel capacity.

These results are identical to the finite length performance results.

Since $r_G = \{\frac{1}{2}, \frac{1}{3}\}$ are commonly in our use-cases, therefore we choose to set the base code rate to $r_L = 2/3$.

Table 11 Average decoding inefficiency ratio as a function of $N1$ and the decoding scheme (IT+RS versus ML), for $K = 1000$ and $r_L = \frac{2}{3}$

r_G	$N1$	(IT+RS) decoding	ML decoding
$\frac{2}{3}$ ($E = 0$)	3	1.06669	1.04225
	5	1.09682	1.00636
	6	1.12217	1.00396
	7	1.14624	1.00250
$\frac{1}{2}$ ($E = 1$)	3	1.10487	1.01627
	5	1.22160	1.00097
	6	1.27825	1.00040
	7	1.32857	1.00009
$\frac{1}{3}$ ($E = 3$)	3	1.17963	1.00668
	5	1.42080	1.00019
	6	1.53045	1.00007
	7	1.62660	1.00001

Table 12 Decoding threshold comparison between (IT+RS) decoding and ML decoding versus $N1$ for different values of r_L and r_G

r_L	Decoding threshold	$N1$	$E = 2$	$E = 3$		
$r_L = \frac{2}{3}$	$\varepsilon^{(IT+RS)}$	3	0.5532	0.6184		
		4	0.5136	0.5753		
		5	0.4744	0.5332		
		6	0.4392	0.4955		
		$\bar{\varepsilon}^{ML}$	3	0.5946	0.6633	
			4	0.5990	0.6661	
	5		0.5998	0.6666		
	6		0.5999	0.6666		
	$r_L = \frac{1}{2}$		$\varepsilon^{(IT+RS)}$	3	0.7236	0.7769
				4	0.7002	0.7530
		5		0.66698	0.7232	
		6		0.6387	0.6927	
$\bar{\varepsilon}^{ML}$		3		0.7403	0.7933	
		4		0.7476	0.7985	
		5	0.7493	0.7996		
		6	0.7498	0.7999		

5 Performance evaluation

In Section 4, we concluded that scheme A, $N1 = 5$, and small number of extra-repair symbols distributed regularly on generalized check nodes are the most appropriate values for hybrid (IT/RS/ML) decoding. Based on this optimal configuration, we now investigate more in details the performance achieved in various situations, for different performance metrics. Then, we compare with the performance achieved with other AL-FEC codes. Finally, we discuss the decoding complexity of GLDPC-Staircase codes and give preliminary decoding throughput results.

5.1 Achieved performance

5.1.1 (IT+RS) versus hybrid (IT/RS/ML) decoding

Let us first quantify the gains made possible by the use of ML decoding. Figure 14 shows the average (over 1000 different codes) decoding inefficiency ratio for various object sizes or equivalently code dimensions K (both are equal here as the object is encoded in a single pass), when $r_G = \frac{1}{2}$. It confirms the major performance gains for all object sizes (e.g., equal to 22 % for $K = 1000$). More remarkable, the performance remains excellent for very small values of K .

5.1.2 Detailed hybrid (IT/RS/ML) decoding inefficiency ratio results

In Fig. 15a, we plot the average (over 1000 different codes) hybrid (IT/RS/ML) decoding inefficiency ratio as a function of the object size for various code rates: $r_G = \frac{1}{3}$

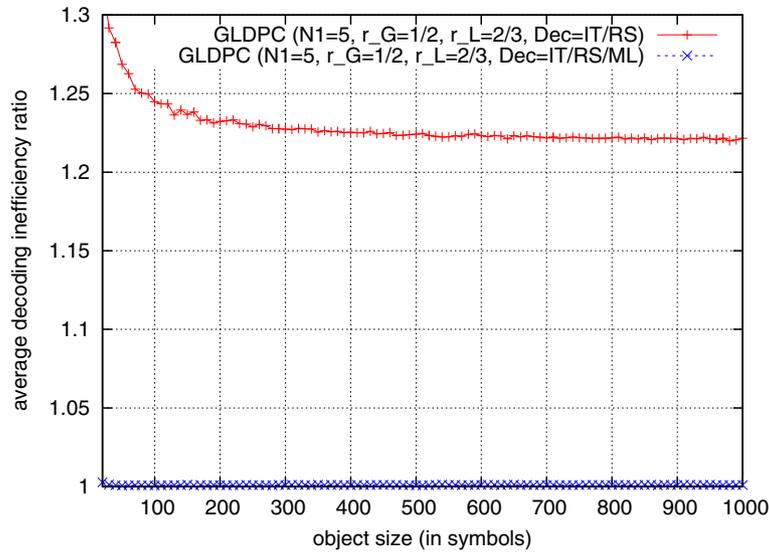


Fig. 14 Comparison of (IT+RS) and (IT/RS/ML) decodings. Average decoding inefficiency ratio as a function of the object size

and versus the number of received symbols in Fig. 17a, when the code rate is $\frac{1}{2}$. Figure 17b shows that the GLDPC-Staircase codes are close to ideal with a very steep slope in the “waterfall” area. In addition, no error floor appears above 10^{-6} decoding failure probability, which is a very good performance, whereas LDPC-Staircase codes exhibit an error floor at 2.10^{-5} decoding failure probability. Figure 17a shows that with GLDPC-Staircase codes, the reception of a few symbols in addition to K allows to quickly reach a decoding failure probability below 10^{-6} . Besides, in order to obtain ML decoding failure probability equals to 10^{-4} , 28 symbols in addition to $K = 1024$ are needed with LDPC-Staircase codes, against 4 symbols with GLDPC-Staircase codes. Therefore, GLDPC-Staircase codes provide major gains with low error floor and a very steep slope compared to LDPC-Staircase codes.

5.2.2 Comparison with another GLDPC code [26]

Let us now compare the correction capabilities with the GLDPC code construction defined by Tanner [3]. This GLDPC code is characterized by an optimal distribution (capacity-approaching) based on hybrid check node structure, which is composed of SPC and (31,21) linear block codes (BCH codes with $d_{\min} = 5$) [26]. For our codes, we use a regular distribution of extra-repair symbols and a base code with $N1 = 5$, $r_L = \frac{2}{3}$ and DD as shown in Table 1).

Table 15 provides a comparison in terms of decoding threshold for rate $\frac{1}{2}$. It shows that our construction method performs the best.

5.2.3 Comparison with Raptor and RaptorQ codes

Finally, we compare with Raptor codes [25] when

- Case 1: $K = 32, N = 128$;
- Case 2: $K = 256, N = 1024$;
- Case 3: $K = 1024, N = 3072$.

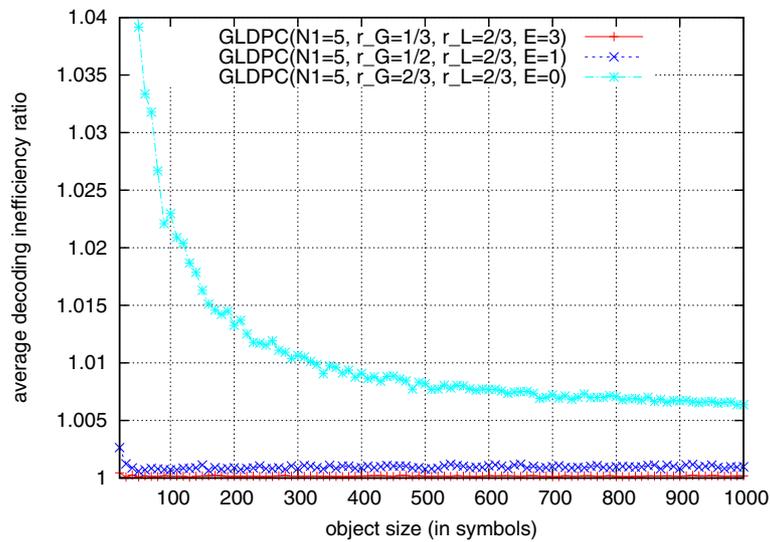
In Fig. 18, we plot the ML decoding failure probability versus decoding overhead for those three cases. For Raptor codes, we used the results provided by Qualcomm for 3GPP in [27]. This figure shows that GLDPC-Staircase codes outperform Raptor codes, no matter the object size. In fact, in case 3, with two symbols in addition to K GLDPC-Staircase achieves a decoding failure probability equal to 1.2×10^{-4} compared to 0.41562 for Raptor codes.

Finally, using the results given in [27], the correction capabilities of our codes are close to those of RaptorQ codes. For instance, in case 3, the 10^{-7} decoding failure probability is achieved with two additional symbols for RaptorQ codes and four symbols in our case.

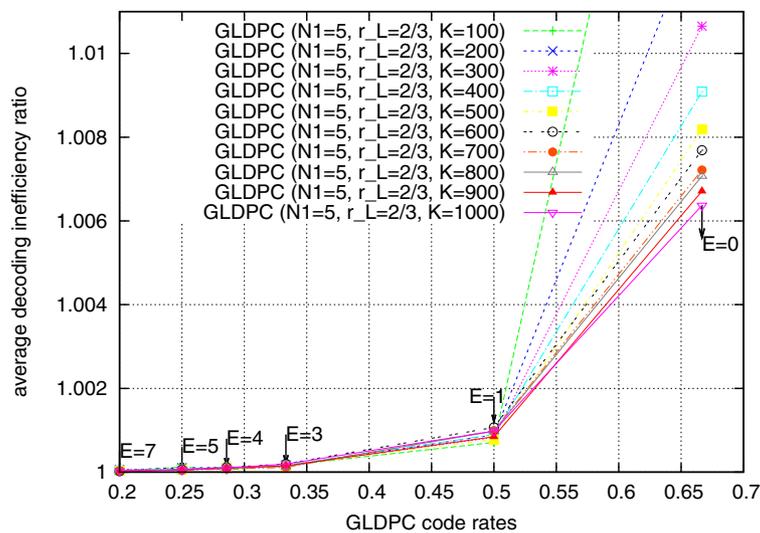
5.3 Hybrid (IT/RS/ML) decoding complexity

5.3.1 Experimental conditions

In order to evaluate the complexity of GLDPC-Staircase codes, we implemented a functional C language codec integrated in the <http://openfec.org> environment, reusing its binary LDPC-Staircase and Hankel-RS over $GF(2^8)$ software codecs. Then, we conducted throughput tests on a Linux desktop, using kernel 2.6.27.41/64 bits, and an Intel Xeon CPU E5410@2.33GHz processor (a single CPU core was used during experiments). Here, we also use $N1 = 5$,



(a)



(b)

Fig. 15 Impacts of the E parameter on decoding performance. Average decoding inefficiency ratio as a function of the object sizes (a) and code rates (b)

$r_L = 2/3$, and a regular distribution of extra-repair symbols.

5.3.2 Results

Hybrid decoding always starts with the (IT+RS) decoder and triggers the ML (binary and/or non-binary) decoder if needed. This typically happens when the channel loss rate is too high for the (IT+RS) decoder to recover all the erased symbols. During ML decoding, recovering source and repair LDPC symbols using binary ML decoding requires only simple XOR operations.

However, extra-repair symbols add finite field operations: multiplying symbols with associated coefficients over $GF(2^8)$ and adding symbols. If finite field addition consists in XORing the two values, finite field multiplications are more complex, requiring in general a log table lookup, an addition operation, and an exponentiation table lookup to determine the result. With $GF(2^8)$, a common optimization with software codecs consists in pre-calculating multiplications and storing the result in a 255×255 table. With this optimization, multiplying two elements of $GF(2^8)$ consists in accessing the right element

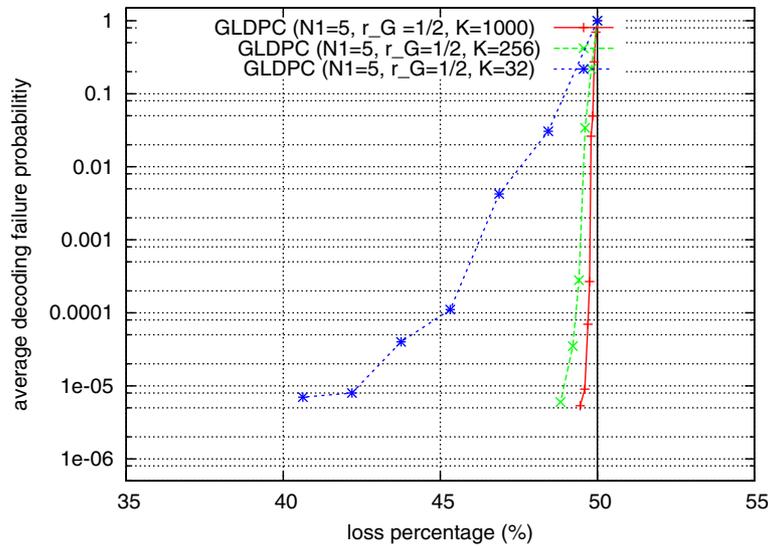


Fig. 16 Hybrid (IT/RS/ML) decoding failure probability of GLDPC-Staircase codes. Performance as a function of the channel loss percentage when $r_G = \frac{1}{2}$ and $K = 32, K = 256, \text{ or } K = 1000$

Table 14 ML decoding failure probability of GLDPC-Staircase codes as a function of the overhead when $r_G = \frac{1}{2}$ and $K = 32, K = 256, \text{ or } K = 1000$

K	Reception overhead	Average decoding failure prob.
32	0	0.0305
	1	4.2×10^{-3}
	2	1.1×10^{-4}
	3	4×10^{-5}
	4	8×10^{-6}
	5	7×10^{-6}
	6	2×10^{-6}
256	0	0.22
	1	0.0351
	2	1.18×10^{-3}
	3	7.39×10^{-4}
	4	4.97×10^{-4}
	5	3.35×10^{-4}
	6	1.37×10^{-4}
1000	0	0.6967
	1	0.2725
	2	0.0494
	3	0.0262
	4	2.68×10^{-4}
	5	6.96×10^{-5}
	6	9×10^{-6}

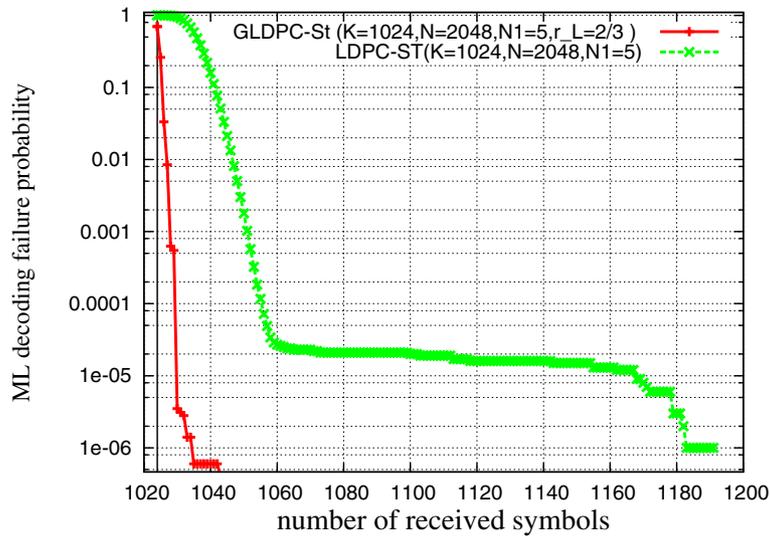
of this pre-calculated table, and multiplying a symbol by a coefficient over $GF(2^8)$ consists in doing this for all the bytes of the symbol.

In Fig. 19, we plot the decoding throughput in megabyte per second as a function of the channel loss percentage. We set $r_G = \frac{1}{2}, E = 1$, and consider an object of size $K = 1000$ symbols, each of them of size 1024 bytes.

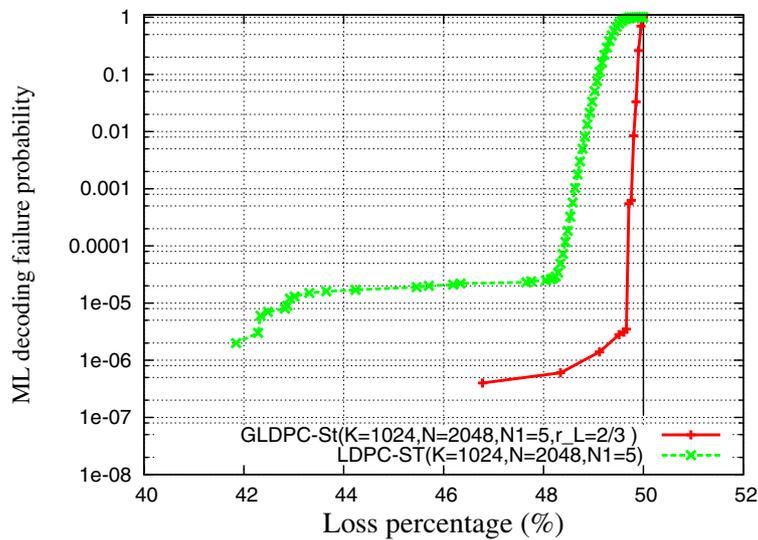
This figure illustrates what we said above: when the channel loss percentage is low (until 35 %), the (IT+RS) decoding is usually sufficient to recover the erased source symbols with speeds around 700 Mb/s. Then, as the channel loss percentage approaches the theoretical limit (i.e., 50 %), the throughput decreases until it stabilizes around 50 Mb/s. This is due to the frequent use of ML decoding. If (IT+RS) decoding is fast, ML decoding remains costly and needs further optimizations. In [28], the structured Gaussian elimination (SGE) method has been successfully applied to LDPC-Staircase codes. An extension of this approach to GLDPC-Staircase codes should reduce the hybrid decoding complexity.

6 Conclusions

This paper provides the fundamentals for the design and analysis of GLDPC-Staircase AL-FEC codes, a class of codes well suited to reliable transmissions and large-scale content distribution, in particular when retransmissions are not possible. They can be viewed as an intelligent way of coupling two complementary codes: on the one hand, the *structured binary* LDPC-Staircase codes, characterized by low encoding/decoding complexities yet good performance for medium to large objects (where RS codes behave poorly); on the other hand, the *non-binary*



(a)



(b)

Fig. 17 LDPC Staircase versus GLDPC-Staircase. ML decoding failure probability comparison for $K = 1024$ and rate $\frac{1}{2}$

RS codes that are ideal codes for small objects (where LDPC codes behave poorly). The coupling is as follows: each SPC check node of LDPC-Staircase (base code) is replaced by an RS code (component code), nodes that are now called “generalized check nodes.” We also define and analyze two variants, schemes A and B, that differ by the nature of the first repair symbol of each generalized check node. Thanks to the generalized check nodes, a large number of RS repair symbols (called *extra-repair symbols*) can be produced on demand, a nice feature for situations where the channel conditions can be worse than expected or to fountain-like content distribution

applications. Very small rates are therefore easily achievable.

First of all, we introduce the use of systematic “quasi” Hankel-RS codes, a forgotten type of RS codes, as a practical way to design GLDPC-Staircase codes (both

Table 15 Decoding threshold of GLDPC-Staircase and another GLDPC construction for rate $\frac{1}{2}$

Code type	Decoding threshold
GLDPC-Staircase, Hankel-RS	0.4993
GLDPC, BCH	0.4967

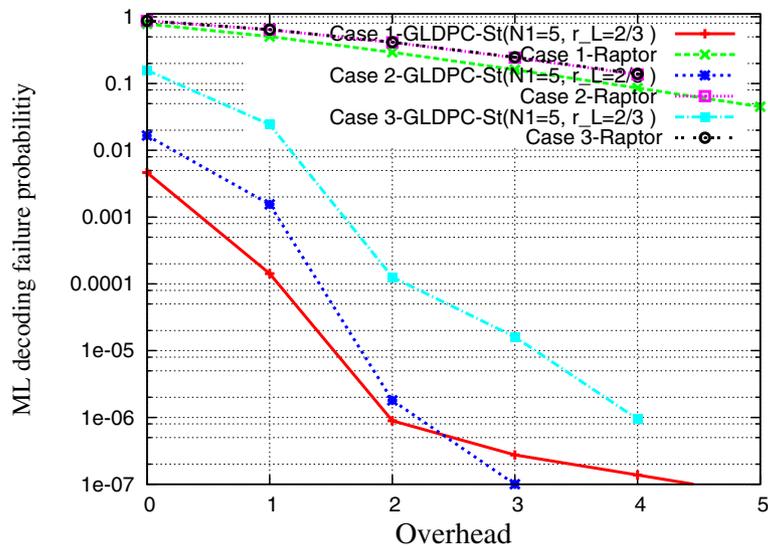


Fig. 18 GLDPC-Staircase versus Raptor. ML decoding failure probability versus overhead in three cases

variants). We show that this class of RS codes features very low construction times, which means that GLDPC-Staircase codes can be generated on the fly, with the exact code dimension and length values. In addition to the basic (IT+RS) decoder of GLDPC-Staircase codes, we introduce a new hybrid (IT/RS/ML) decoder, and thanks to a joint use of IT, RS, and binary and non-binary ML decoding, it achieves ML decoding performance at a lower complexity. This is confirmed by preliminary decoding throughput tests, even if further optimizations are needed for the ML decoder.

Then, we detail the asymptotic analysis method for GLDPC-Staircase codes under both (IT+RS) and ML decoding, using the EXIT and DE tools. It allows us to investigate their decoding convergence and gap to the theoretical limit. First, we derive DE equations under (IT+RS) decoding for the two variants, schemes A and B. This technique is then combined with EXIT functions by generalizing the area theorem (initially proposed for LDPC codes) to our codes, and we determine the upper bound on the ML decoding threshold. This analysis shows that GLDPC-Staircase codes under ML decoding

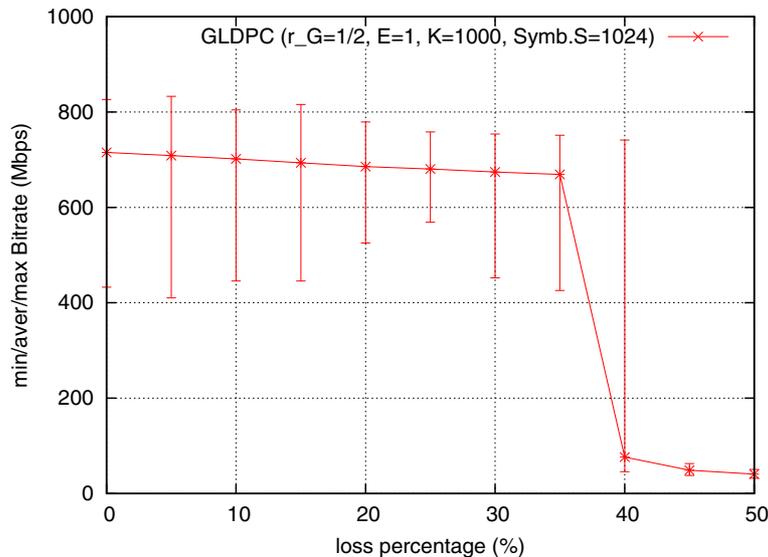


Fig. 19 Throughput of hybrid (IT/RS/ML) decoding. GLDPC-Staircase codes with $r_G = \frac{1}{2}, E = 1$

are quite close to the theoretical limit. Moreover, using this method along with a finite length analysis, we discuss the impacts of the code structure. We show that the “dual identity” repair symbols of scheme A improve the decoding correction power of each generalized check node (the impact of stopping sets to (IT+RS) decoding vanishes) which accelerates the (IT+RS) decoding convergence. Since this is done without negatively impacting ML decoding performance, we conclude that scheme A is the most appropriate code construction approach. Our analyses also lead us to conclude that $N1 = 5$ and a small number of extra-repair symbols distributed regularly over the generalized check nodes represent the best configuration for hybrid (IT/RS/ML) decoding.

Finally, using finite length analyses and the above optimal configuration, we show that these codes exhibit exceptional erasure recovery capabilities with a memory-less channel. More precisely, they show very small decoding overheads, close to that of ideal codes, low decoding failure probabilities, low error floors, and steep waterfall regions. These results are achieved no matter the object size, which is the main problem of most AL-FEC codes (e.g., LDPC-Staircase, RS, and Raptor codes). From this point of view, GLDPC-Staircase codes outperform the LDPC-Staircase, Raptor, and another construction of GLDPC codes [8, 26]. Their correction capabilities are close to that of RaptorQ codes.

All these results make GLDPC-Staircase codes a ubiquitous class of AL-FEC codes for the erasure channel. They are very well suited to streaming applications where encoding is performed on small amounts of data in order to satisfy real-time constraints. But they are also appropriate for bulk data transfer applications where the encoding is performed on large amounts of data (ideally a single block encompassing the whole file) where it is only limited by the practical aspects (available memory and CPU during decoding).

Endnotes

¹This matrix is divided into two parts and has the form $(H_1|H_2)$. The $M_L \times K$ left-hand side part, H_1 , defines the placements of source symbols in equations (rows). It is created in a regular way in order to have constant column and row degrees. More precisely, each column of H_1 is of degree $N1$, which is an input parameter during the LDPC-Staircase code creation [12]. The $M_L \times M_L$ right-hand side part, H_2 , defines in which equations the repair symbols are involved and features a staircase (i.e. double diagonal) structure.

²As explained previously, with scheme A, the k_m symbols consist of x plus p_{m-1} (if $m > 1$). With scheme B, they consist of x plus p_{m-1} (if $m > 1$) and p_m .

³We note that hybrid (IT/RS/ML) decoding and ML decoding have the same correction capabilities but they are different at decoding complexity level. Thus, we mention “ML decoding” to refer the correction capabilities obtained by hybrid decoding.

⁴It is equivalent to the order of packet loss.

⁵It is the number of required symbols over K to succeed the decoding

⁶It is the ratio between the number of symbols needed for decoding to succeed over the number of source symbols.

⁷It is the probability that at least one erased source symbol is not recovered

Acknowledgements

This work was supported by the ANR-09-VERS-019-02 grant (ARSSO project) and by the INRIA—Alcatel Lucent Bell Labs joint laboratory.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Inria, PRIVATICS research Team, Grenoble, France. ²Alcatel-Lucent Bell labs, Paris, France.

Received: 7 October 2015 Accepted: 18 June 2016

Published online: 05 September 2016

References

1. R Gallager, Low Density Parity Check codes. *IRE Trans. Inform. Theory*. **8**, 21–28 (1962)
2. D MacKay, R Neal, Near Shannon limit performance of low density parity check codes. *IET Electron. Lett.* **33**(6), 457–458 (1997)
3. R Tanner, A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*. **27**(5), 533–547 (1981)
4. J Boutros, O Pothier, G Zéor, Generalized Low Density (Tanner) Codes. *IEEE Int. Conf. Commun. (ICC)*, 441–445 (1999)
5. M Lentmaier, K Zigangirov, On Generalized Low-Density Parity-Check codes based on Hamming component codes. *IEEE Trans. Commun.* **3**(8), 248–250 (1999)
6. G Yue, L Ping, X Wang, Generalized low-density parity-check codes based on Hadamard constraints. *IEEE Trans. Inform. Theory*. **53**(3), 1058–1079 (2007)
7. J Chen, R Tanner, A hybrid coding scheme for the Gilbert-Elliott channel. *IEEE Trans. Commun.* **54**(10), 1787–1796 (2006)
8. N Miladinovic, M Fossorier, Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels. *IEEE Global Telecommun. Conf. (GLOBECOM)*, 1239–1244 (2005)
9. I Djordjevic, O Milenkovic, B Vasic, Generalized Low-Density Parity-Check codes for optical communication systems. *Lightwave Technol.* **23**(5), 1939–1946 (2005)
10. Y Wang, M Fossorier, Doubly Generalized LDPC codes. *IEEE Int. Symp. Inform. Theory*, 669–673 (2006)
11. E Paolini, D Wilab, M Fossorier, M Chiani, in *Proceedings of Allerton Conference on Communications, Control and Computing*. Analysis of doublygeneralized ldpc codes with random component codes for the binary erasure channel, (2006)
12. V Roca, C Neumann, D Furodet, Low Density Parity Check (LDPC) staircase and triangle forward error correction (FEC) schemes (2008). IETF Request for Comments, RFC 5170
13. M Cunche, V Savin, V Roca, G Kraidy, A Soro, J Lacan, Low-rate coding using incremental redundancy for GLDPC codes. In: *IEEE Int. Workshop on Satellite and Space Communications* (2008)

14. F Mattoussi, V Roca, B Sayadi, Design of small rate, close to ideal, GLDPC-Staircase AL-FEC codes for the erasure channel. *IEEE Global Commun. Conf. (GLOBECOM)*, 2143–2149 (2012)
15. F Mattoussi, V Roca, B Sayadi, Good coupling between LDPC-staircase and Reed-Solomon for the design of GLDPC codes for the erasure channel. *IEEE Wireless Commun. Netw. Conf. (WCNC)*, 1528–1533 (2013)
16. F Mattoussi, V Roca, B Sayadi, Complexity comparison of the use of Vandermonde versus Hankel matrices to build systematic MDS Reed Solomon codes. *IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 344–348 (2012)
17. E Paolini, G Liva, B Matuz, M Chiani, Generalized IRA erasure correcting codes for hybrid iterative/maximum likelihood decoding. *IEEE Commun. Lett.* **12**(6), 450–452 (2008)
18. M Cunche, V Roca, Optimizing the error recovery capabilities of LDPC-Staircase codes featuring a Gaussian elimination decoding scheme. In *10th IEEE International Workshop on Signal Processing for Space Communications (SPSC 2008)* (2008)
19. TJ Richardson, RL Urbanke, The capacity of Low-Density Parity-Check codes under message-passing decoding. *IEEE Trans. Inform. Theory.* **47**(2), 599–618 (2001)
20. M Luby, M Mitzenmacher, M Shokrollahi, D Spielman, V Stemann, Practical loss-resilient codes. *29th Annual ACM Symposium on the Theory of Computing* (1997)
21. S ten Brink, Convergence of iterative decoding. *IEEE Electron. Lett.* **35**(10), 806–808 (1999)
22. C Mésson, A Montanari, R Urbanke, in *IEEE Int. Symposium on Information Theory (ISIT)*. Maxwell's construction: The hidden bridge between maximum-likelihood and iterative decoding, (2004)
23. C Messon, Conservation laws for coding. PhD thesis, Ecole Polytechnique Federale de Lausanne (EPFL) (2006)
24. M Cunche, Codes al-fec hautes performances pour les canaux défacements : variations autour des codes ldpc (2010). PhD thesis, Université Joseph Fourier, Grenoble
25. M Luby, A Shokrollahi, M Watson, T Stockhammer, Raptor forward error correction scheme for object delivery. IETF Request for Comments, RFC 5053 (2007)
26. E Paolini, Iterative decoding methods based on low-density graphs. PhD thesis, Università degli Studi di Bologna (2007)
27. EMM-EFEC, Selection of the FEC. Document S4-121367, 3GPP TSG-SA4 meeting 71, Bratislava, Slovakia (2012). http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_71/Docs/S4-121367.zip
28. V Roca, M Cunche, C Thienot, J Detchart, J Lacan, in *9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Rs + ldpc-staircase codes for the erasure channel: Standards, usage and performance, (2013)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
